VŠB - Technická univerzita Ostrava

Fakulta elektrotechniky a informatiky

DIPLOMOVÁ PRÁCE

2013 Petr Kriegisch

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Extrakce informací z webových diskuzních fór

Information Extraction from Web Forums

2013 Petr Kriegisch

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Petr Kriegisch**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: Extrakce informací z webových diskuzích fór
Information Extraction from Web Forums

Zásady pro vypracování:

Cílem práce je provedení průzkumu existujících přístupů, návrh a implementace vybrané nebo vlastní metody a aplikačního prostředí pro experimenty.

1. Průzkum a popis existujících přístupů.
2. Návrh a implementace vybrané nebo vlastní metody.
3. Návrh a implementace počítačové aplikace pro provádění experimentů.
4. Návrh, realizace a hodnocení experimentů.

Seznam doporučené odborné literatury:

[1] Bing Liu. Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications). Springer 2011.
Dále dle pokynů vedoucího diplomové práce.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Mgr. Miloš Kudělka, Ph.D.**

Datum zadání: 16.11.2012
Datum odevzdání: 07.05.2013

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Poděkování

Na tomto místě bych rád poděkoval vedoucímu mé diplomové práce, panu Mgr. Miloši Kudělkovi Ph.D, který mi poskytoval rady a byl mi nápomocen během psaní diplomové práce.

Prohlášení

Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava.

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 12. prosince 2012 .....

Abstrakt

Diplomová práce se zabývá prozkoumáním již existujících technik pro získávání dat z webových diskuzních fór, navrhnutí techniky mé, která je implementována v jazyce Java a má sloužit k detekci takového fóra a následné extrakci relevantních dat z příspěvku.

Klíčová slova

Fórum, CSS, HTML, Příspěvek, Java, JSoup

Abstract

This thesis is concerned with examining the existing techniques for extracting data from web discussion forums and devising my technique, which is implemented in Java and is used to detect this forum and subsequent extraction of relevant data from the post.

Keywords

Topic, CSS, HTML, Post, Java, JSoup

Seznam použitých symbolů a zkratek

HTTP – HyperText Transfer Protocol – internetový protokol

Parser – Analyzátor textu

URL – UniformResourceLocator - Přesná identifikace dokumentu na internetu

Wordlist – Seznam slov

Obsah

1. Úvod

Extrakce dat z webových diskuzních fór je důležitá z hlediska vytvoření automatizovaného procesu, který na základě odkazu URL zjistí, zda je daná stránka vláknem webového diskuzního fóra, (dále jen fóra) či nikoliv. Na základě tohoto vyhodnocení bude dále extrahovat další informace, které jsou pro nás zajímavé. Pro lidské oko je takové posouzení triviální oproti strojovému zpracování. Z tohoto důvodu je vhodné vytvořit metodu, která bude schopna detekovat prvky, které vlákno fóra musí splňovat.

K tomu je zapotřebí přečíst a zpracovat zdrojový kód v HTML, který každá webová stránka obsahuje a podle předem stanovených pravidel určit, zda stránka pravidla splňuje či nikoliv.

S rozmachem internetu je spousta užitečných dat uložena právě v těchto fórech. Pro většinu lidí je samozřejmě jednodušší v teple domova sednout k internetu a potřebné novinky, inzeráty, produkty, zprávy či jiné informace hledat na internetu, nežli nakupovat časopisy, noviny, inzertní časopisy nebo katalogy se zbožím.

Jenomže nemůžeme zajistit a také by to bylo nevhodné z důvodu odlišnosti webových stránek, aby každý autor, který webovou stránku vytvořil, použil předem dané pravidla a zajistil tak, aby všechny webové diskuzní fóra, která jsou podmnožinou webových stránek, měly stejnou strukturu a bylo by tedy možné použít jednotnou metodu k získání dat. K tomu účelu je zapotřebí jednotlivé struktury zdrojových kódu od sebe odlišit a popřípadě sloučit podobné modely webových stránek pro další zpracování.

Cílem diplomové práce bude tedy prozkoumat již existující metody, které pro tento účel slouží a vytvořit vlastní program, kde bude použita jedna z již existujících metod nebo vytvořena metoda vlastní.

1.1 Webová stránka

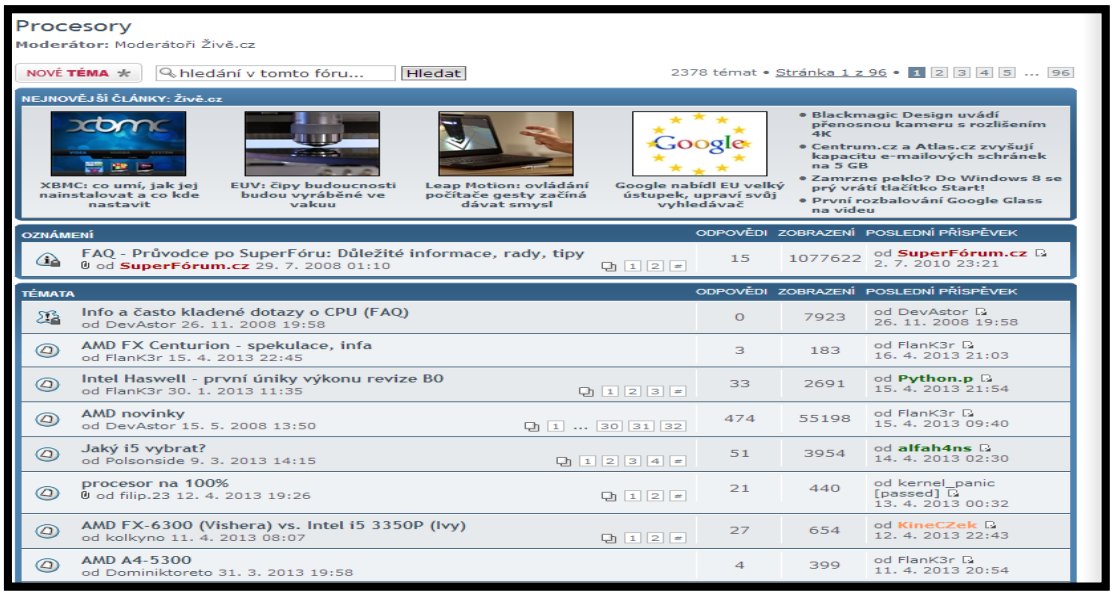
Webová stránka je dokument, který je fyzicky umístěn v rámci celosvětové WWW (World Wide Web) sítě kdekoliv na webovém serveru a slouží k zobrazování informací uživateli, který k dokumentu přistoupí prostřednictvím této sítě. Za pomocí protokolu HTTP komunikuje s klientem, který si žádá danou webovou stránku a za pomocí internetového prohlížeče ji zobrazí. Klient zastupuje funkci uživatele. WWW síť je propojena pomocí hypertextových odkazů, kde každý jeden zastupuje právě jednu webovou stránku. Takováto webová stránka je popsaná pomocí jazyku HTML či XHTML, který určuje, jak bude vypadat a co bude obsahovat.

Webové stránky lze rozdělit na statické a dynamické. Statické obsahují pouze text, který se mění jen za pomocí přepisování zdrojového kódu. Oproti tomu dynamické stránky jsou přetvářeny za pomocí samotné webové stránky, kde může klient napsat například svůj názor či dotaz za pomocí formuláře a ten se následně zobrazí na webové stránce. Příkladem může být webové diskuzní fórum. Užívá se zde i databází a skriptovacích programovacích jazyků. Z pohledu klienta zde není potřeba žádný dodatečný software.

V rámci webových stránek lze uvažovat ještě další rozdělení, a to v kombinaci s databází, která slouží jako uložiště pro data. Zde existuje jedna nebo více webových stránek, které obsahují seznam produktů nebo příspěvků, zobrazující jejich titulek. Vzhledem k tomu, že diplomová práce je zaměřena na webová diskuzní fóra, budeme tedy brát jako vzor právě tento druh webových stránek obsahující příspěvky. Druhou nedílnou součástí k těmto webovým stránkám existuje mnoho dalších, které zobrazují detailní pohled na jednotlivé položky ze seznamu, čili příspěvky. Ty jsou již zobrazeny za pomocí šablony, která funguje jako předpis vzhledu a struktury pro všechny ostatní. Vzhled se u jednotlivých příspěvků neliší, pouze data, která jsou načítána z databáze.

Rozdíl mezi webovou stránkou obsahující seznam jednotlivých příspěvků a webovou stránkou, která obsahuje již detailní pohled na jeden příspěvek je zobrazen na Obrázek 1 a Obrázek 2.

V diplomové práci se zaměřím na práci s druhou variantou webové stránky, a to konkrétně na extrakci informací z těchto detailnějších pohledů na příspěvky.



Obrázek 1 - Přehledová webová stránka

Obrázek 2 - Detailní webová stránka - příspěvek

U Obrázek 1 jsou jednotlivé řádky reprezentovány jako dílčí záznamy, které spolu nemusejí mít nic společného. Je to tedy jeden záznam za druhým. Na Obrázek 2 je zobrazen jeden konkrétní včetně detailního popisu týkající se pouze jednoho záznamu z přehledové webové stránky. V našem případě seznam odpovědí, které jsou reakcemi na první příspěvek. Více o struktuře webového diskuzního fóra bude napsáno v další kapitole.

1.2 Webové diskuzní fórum

Webové diskuzní fórum je speciálním typem webové stránky, kde uživatelé vytvářejí tzv. příspěvky, do nichž vkládají své dotazy či zkušenosti a ostatní uživatelé k tomu připisují své názory a nápady. Každý takový uživatel vystupuje pod svou přezdívkou, kterou si sám zvolí a u každého jeho příspěvku je tohle jméno zobrazeno. Počet zpráv není nijak omezen a uživatelé tak mohou spolu diskutovat neomezeně.

Diskuzní fóra jsou organizována do různých skupin a to podle tématu, do kterého daný příspěvek zapadá. Každá skupina obsahuje již zmíněné příspěvky. Příspěvek má svůj titulek, který by měl vystihnout co nejpřesněji, o čem se zde bude diskutovat. Dalším povinným atributem příspěvku je jeho autor a datum, kdy byl příspěvek vložen. Velice často bývá omezen pouze počet příspěvků zobrazených na jednu stránku např. 20 příspěvků a další jsou zobrazeny na následující stránce. Struktura je zde většinou chronologická a nejnovější příspěvky jsou uloženy na konec stránky. Mezi jednotlivými stránkami lze libovolně listovat.

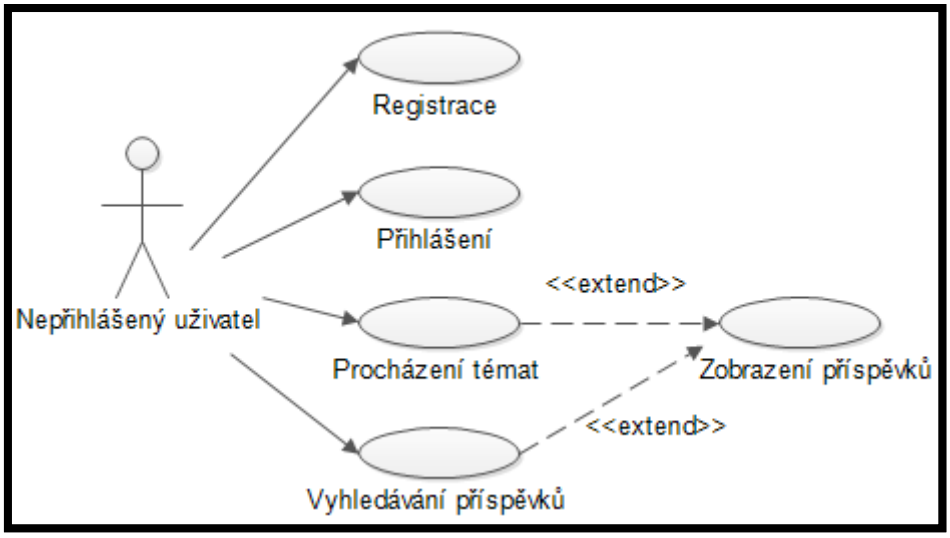
Oproti klasickému chatu se webové diskuzní fórum liší v tom, že uživatelé, kteří spolu za pomoci příspěvku komunikují, nemusejí být online a mohou tak odpovídat kdykoliv nezávisle na čase.

Diskuzní fórum může být součástí webového portálu, který nabízí k prodeji produkty a v rámci téhle webové prezentace je i diskuzní fórum, kde uživatelé spolu diskutují o produktech. Často ale fóra nejsou vytvořena za konkrétním účelem či tématem, ale obsahují průřez všemi různými tématy. Počínaje nabídkou práce, diskutování o hudbě, filmech až po témata týkajících se mobilních telefonů a automobilů. Na jakékoliv téma se dá jistě najít fórum.

Vzhledem k tomu, že se jedná o zobrazování webových stránek, není nutné instalovat žádný další podpůrný software.

Webová diskuzní fóra rozlišují ve většině případů tři hlavní role. A to přihlášeného uživatele, nepřihlášeného uživatele a moderátora. Hlavním rozdíl je v možnosti prováděných aktivit v rámci webového diskuzního fóra. Moderátor má na starost moderování jednotlivých příspěvků a to ve smyslu dodržovaní pravidel, ke kterým se uživatel při registraci zavázal. Mezi hlavní takováto pravidla patří např. dodržování spisovného jazyka, vkládání příspěvků do patřičných témat, nepublikování jakýchkoliv reklam či vkládání příspěvků porušujících zákony. Z toho vyplývá, že moderátor má kompletní přístup ke všem jemu náležícím příspěvkům, se kterými může provádět editace, mazání či přesunování. Speciální rolí je administrátor, který nemá na starost jednotlivé příspěvky tak jako moderátor, ale např. správu uživatelů nacházejících se v systému. Samozřejmě má také možnost kompletní správy příspěvků a vychází tak z role moderátora ovšem s právy na všechny příspěvky.

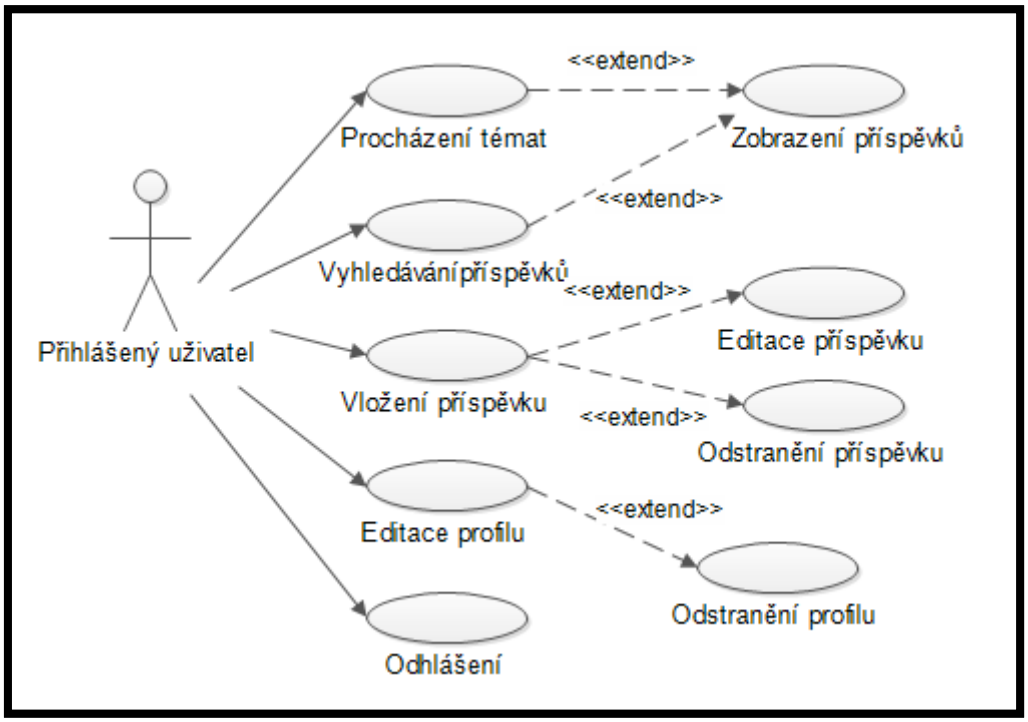
Činnosti nepřihlášeného uživatele je zobrazena na Obrázek 3 a aktivity přihlášeného uživatele na Obrázek 3.



Obrázek 3 - Nepřihlášený uživatel

Tento případ užití není přesným odrazem aktivit, prováděných ve všech fórech. Každý takový portál poskytuje spoustu dalších funkcí, jež může uživatel provádět. Na Obrázek 3 jsou zobrazeny pouze základní aktivity, které by měli všeobecně poskytovat autoři webových stránek zaměřených na vkládání příspěvků a procházení reakcí na ně.

Nepřihlášený uživatel má možnost registrace, čímž si zpřístupní ve většině případů vkládaní nových příspěvků či reakcí na ně. Existují však i webové diskuzní fóra, kde není registrace nutná a uživatel tak vystupuje anonymně. Lze tak vkládat příspěvky bez registrace. Pokud je již uživatel registrován tak se přihlásí. Dále, co může nepřihlášený uživatel provádět je procházení témat, které se na webové stránce vyskytují. I to není vždy pravidlem, jelikož někteří autoři webových stránek si vynucují přihlášení, aby bylo možné témata zobrazit. Pokud se však zobrazí přehled témat i bez přihlášení, může uživatel číst jednotlivé příspěvky nacházející se v daném tématu. Poslední důležitou funkčností je samotné vyhledávání příspěvků či klíčových slov pokud autor webového portálu nezpřístupní tuhle funkčnost pouze pro přihlášené uživatele.



Obrázek 4 - Přihlášený uživatel

Přihlášený uživatel má oproti nepřihlášenému uživateli možnost vždy vkládat nové příspěvky a posléze je i editovat či mazat. Další, ale již méně užívanou funkcí, kterou uživatel může vykonávat je editace vlastního profilu. Vzhledem k tomu, že většinou krom hesla nebo emailové adresy není co upravovat, je to méně využívána funkcionalita.

2. Extrahování dat

2.1 Obecný přístup

Naším cílem je z webové stránky, která splňuje podmínky pro webové diskuzní fórum získat data, která jsou pro nás zajímavá. Různé reklamy či jiné informace netýkající se daného tématu jsou pro nás nezajímavá a můžeme je tedy vynechat pro další zpracování.

K tomu, abychom mohli identifikovat co je pro nás důležité, je potřeba prozkoumat kompletní webovou stránku a za pomocí tagů HTML nalézt relevantní úseky, ve kterých jsou uloženy pro nás užitečné informace.

Zpracování webové stránky, která byla označena jako webové diskuzní fórum, probíhá tedy ve třech krocích:

- Nalezení sekce v kódu HTML, která je označena jako příspěvek. Takových příspěvků může být na stránce několik. Vzhledem k tomu, že všechny příspěvky mají stejnou strukturu, může jejich zpracování probíhat stejným způsobem.
- Po nalezení sekce příspěvku pomocí extrakčních pravidel získáme samotný text, který je zbaven všech HTML tagů.
- Výpis získaného textu na výstup či další manipulace s daty ve smyslu uložení do databáze nebo do strukturovaného dokumentu, jako třeba XML.

2.2 Rozdělení přístupu

Každý autor webové stránky má při tvorbě, nežli začne, dvě možnosti. Buďto využije jednu ze šablon, která má jasně danou strukturu nebo vytvoří svou vlastní, kde si zajistí svou odlišnost od jiných podobných webových stránek. Samozřejmě obě možnosti mají své výhody i nevýhody.

Použitím šablony je tvorba takovéto stránky mnohem rychlejší a snadnější, nežli vytvářet úplně něco nového. I orientace pro následné návštěvníky bude jistě příjemnější, jelikož již něco podobného určitě viděli. V neposlední řadě to má nespornou výhodu v tom, že webové vyhledávače, jako např. Google snadněji prozkoumají tuhle webovou stránku a můžou ji tak zahrnout mezi výsledky vyhledání uživatelem. A právě i pro extrakční programy je známá struktura snadněji prozkoumatelná. Naproti tomu nevýhodou může být pro autora to, že se jeho stránka podobá ostatním a nijak se tedy od nich neodliší, což nemusí přilákat větší množství lidi právě kvůli vzhledu, který je tuctový.

Při vytvoření vlastní struktury webové stránky si dá autor jistě více práce, ale na druhou stranu si zajistí určitou odlišnost od těch ostatních a není tak limitován něčím, co vytvořil někdo jiný.

Vzniká tedy problém, jak napsat software, který se vypořádá právě s co největším počtem webových diskuzních fór. Pokud by byla struktura všech takovýchto webových stránek stejná, stačilo by napsat program tak, aby podle předem známých parametrů prozkoumal právě ty části HTML kódu, ve kterých by věděl, že se uchovávají ona data, která chceme extrahovat. Takový program bude sice fungovat, ale pouze na určité procento webových stránek, které jsou napsány právě pomocí téhle šablony. Pokud bychom narazili na jinou, která má odlišnou strukturu, software již nebude vědět, kde se nachází požadované data a nebude tak schopen je extrahovat.

Řešením tohoto problému je napsat program co nejobecněji a vyhledávat tak ve struktuře webové stránky ty prvky, které se často opakují a pomocí zkoumání zdrojových kódu náhodně vybraných fór nalézt skryté šablony, podle kterých lze usuzovat, že právě v těchto částech se příspěvek ukrývá.

2.3 Extrakční techniky

Nástroj, který slouží k samotné extrakci informací z webových stránek, se nazývá Wrapper, jak uvedl [**1**]. Z webových stránek získává data, která jsou uložena v různých šablonách a cílem wrapperu je tedy nalézt za pomocí extrakčních pravidel tyhle data.

V současné době existují společnosti, které se zabývají vytvářením wrapperů a to z důvodu získávání dat z různých zdrojů a jejich seskupením tak co nejvíce podobných dat dohromady. Může tím být zdroj pro prodej online produktů či prohledávání webových diskuzních fór. Na problému ohledně extrakce dat se začalo pracovat již v polovině 90. let a existují 3 hlavní přístupy:

- **Manuální** – Programátor na základě zdrojového kódu vybrané webové stránky nalezne struktury, ve kterých se objevují data, které se mají extrahovat a podle toho napíše vlastní wrapper, který dokáže podle nastavených pravidel získat z webové stránky požadované informace. Tento přístup je však omezen pouze na malé množství webových stránek, protože neexistuje obecnější metoda extrakčních pravidel pracující pro odlišnější webové stránky.
- **Poloautomatický –** Je to tzv. „učení s učitelem“, kde se na základě předem vybrané kolekce webových stránek a jejich dat vytvoří extrakční pravidla. Tyto pravidla jsou pak využívána při prozkoumávání dalších webových stránek, které obsahují podobnou strukturu.

- **Automatický –** Tzv. „učení bez učitele“, kde nejsou určena žádná pravidla, která by byla odvozena na základě předchozích zkušeností. Wrapper sám odhalí na webové stránce strukturu, ve které by se mohla extrahovaná data nacházet. Není tedy potřeba ručně vytvářet sady pravidel, za jejichž pomoci bychom nalezli požadovaná data. Tím se tahle metoda stává velice obecnou a lze tak obsáhnout veliký počet různých webových stránek.

V téhle diplomové práci se budu zabývat druhou metodou, tedy poloautomatickým přístupem, který je také označován jako „učení s učitelem“.

2.3.1 Učení s učitelem

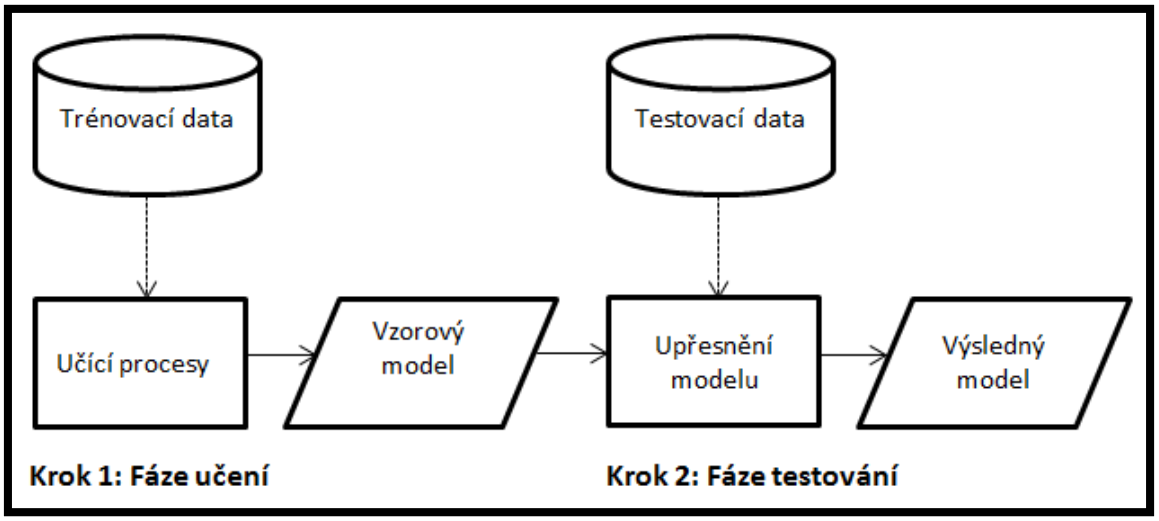
Metoda „učení s učitelem“ (Supervised learning) je založena na stanovených pravidlech, která jsou naučena na předem známých trénovacích datech. Na základě vstupu požadujeme výstup a k tomu potřebujeme vytvořit sadu pravidel, které budou splňovat požadavky pro výstup. Pomocí funkcí stanovíme extrakční pravidla pro wrappery, které slouží k transformaci dat ze vstupu na výstup. Určování pravidel se provádí ručně a pozornost věnujeme pouze objektům, které chceme extrahovat. To nám umožní vybrat pouze relevantní informace. Takto naučená pravidla se použijí pro zpracování webových stránek, které mají stejnou či velice podobnou strukturu. Nelze tím bohužel zajistit, aby všechna tato pravidla byla aplikovatelná na každou webovou stránku, jelikož struktura webových stránek je odlišná.

Vstupem může být webová stránka a výstupem data, která chceme ze stránky extrahovat. Například zadáme-li odkaz na webovou stránku www.seznam.cz a na výstupu chceme zobrazit seznam všech odkazů, které se na stránce vyskytují. Tohle je práce wrapperu.

V metodě „učení s učitelem“ můžeme hledat analogii v reálném světě, kde se člověk na základě předem naučených a získaných zkušeností snaží aplikovat ty nejlepší dovednosti pro situace ve svém životě.

Může se nám to zdát jako věc samozřejmá, ale pro strojové zpracování je to velice obtížný proces, jelikož stroje nemají „zkušenosti“. I tak ale můžeme pomocí metody „učení s učitelem“ dosáhnout aspoň částečného přiblížení k lidskému učení.

Celý proces můžeme znázornit pomocí procesního schématu na Obrázek 5, který ve své práci publikoval Bing Liu [**1**].



Obrázek 5 - Učení s učitelem

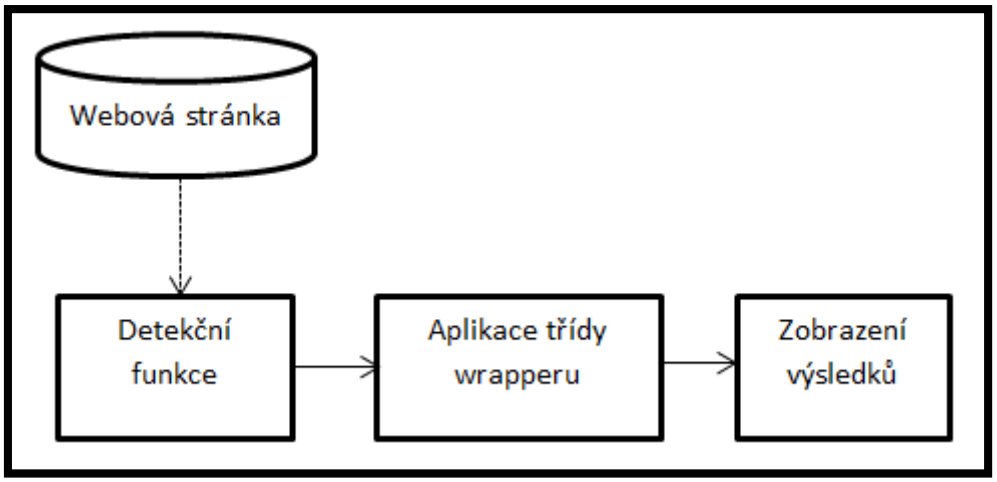
Jako „trénovací data“ zde poslouží množina webových stránek, kde se vyskytují informace, o kterých víme, že budou cílem našeho extrahování. Následujícím krokem jsou „učící procesy“, kde se snažíme objevit v trénovacích datech podobné vazby či struktury, které by mohli posloužit jako vzor pro extrahování informací z webových stránek. Jak již bylo řečeno, nelze najít nějakou univerzální šablonu, která by posloužila pro všechny webové stránky a tím zajistit nalezení funkcí pro 100% úspěšnost v extrahování informací. Tahle fáze je nejnáročnější z celého procesu.

Tímto procházením trénovacích dat a hledáním oněch struktur můžeme vytvořit „vzorové modely“, které by mohly posloužit pro wrapper jako vzory pro extrakci informací z většího množství webových stránek. Následuje fáze testování vzorových modelů, kde zkoušíme úspěšnost extrakce pro různé webové stránky. Jistě nalezneme spoustu odchylek, které způsobí, že naše vzorové modely nebudou fungovat pro větší množství webových stránek a ve fázi „upřesnění“ uděláme drobné úpravy. Pro radikálnější úpravy je vhodné vytvořit nové modely, než upravovat ty stávající. Po dokončení úprav nám vzniknou hotové modely, které dále slouží pro wrappery jako funkce, které transformují vstupní data na výstupní.

3. Wrappery

Wrapper je program, který je vytvořen za účelem extrakce informací z webových stránek, které obsahují určitou strukturu danou jejím zdrojovým kódem napsaným za pomocí jazyka HTML.

Každý takový program je definován sadou funkcí, které provádějí samotnou detekci relevantních úseků zdrojového kódu a posléze extrakci informací, uložených v oněch sekcích. Výsledkem práce wrapperu jsou data, které byly námi označeny jako důležité a za jejichž účelem byl wrapper vytvořen. Jak vypadá práce wrapperu je zobrazeno na Obrázek 6.



Obrázek 6 – Wrapper

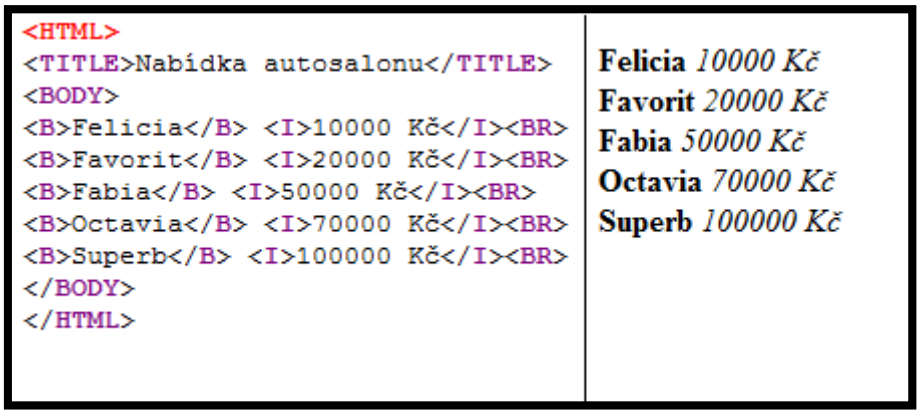
Wrappery lze rozdělit do dvou různých skupin, které se od sebe liší způsobem zpracování dokumentu ve smyslu zpracování zdrojového kódu. Blíže si je popíšeme v následující části, která je zaměřena hlavně na řetězcové wrappery a jejich rozdílné třídy použití.

3.1 Řetězcové wrappery

Samotné wrappery jsou rozděleny do několika tříd, jak popsal ve své práci Matthew Rowe [**2**], které popisují jak ze zdrojového kódu HTML extrahovat důležité informace. Na základě tagů, ze kterých je HTML kód sestaven, se nacházejí požadovaná data. Tyto wrappery pracují se samotným zdrojovým kódem jako s řetězci, čili textem samotným.

3.1.1 Třída LR

Třída LR, dle [**2**] využívá mechanizmu „z leva doprava“ k extrahování informací na základě pravidel, kde každé pravidlo je složeno z levého (L) a pravého (R) ohraničujícího tagu. Uvnitř těchto tagů se nacházejí požadovaná data. Program tedy prohledává zdrojový kód, dokud nenalezne levou značku. Po jejím nalezení extrahuje text, dokud nenalezne pravou značku, která značí konec.



Obrázek 7- Třída LR

Pokud tedy budeme brát jako dvojici parametrů L a R tagy <B> </B> a <I></I> podle Obrázek 7, bude extrahovaný text obsahovat vždy dvojici: „Název automobilu“ a jeho „cena“. Žádné další data na výstupu nebudou. Tahle metoda je využívána jako základ v ostatních třídách řetězcových wrapperů a k parametrům L a R jsou přidávány další, které upřesňují výsledky práce wrapperu.

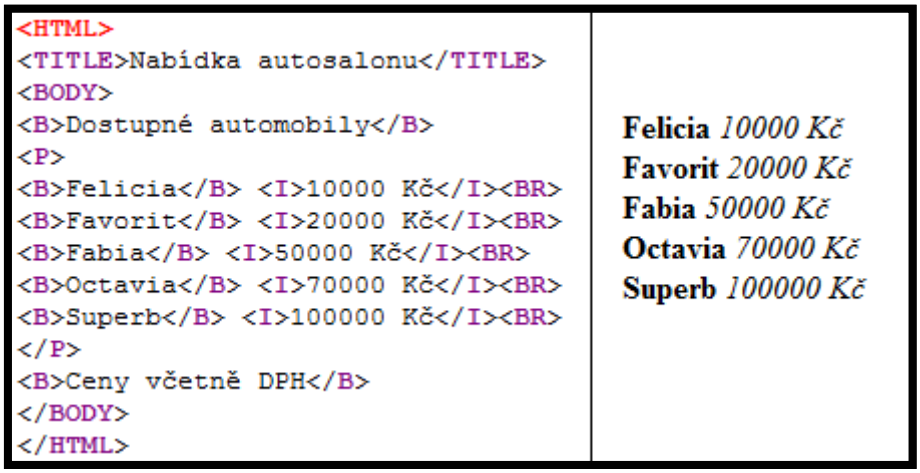
3.1.2 Třída HLRT

Třída HLRT, dle [**2**] vychází ze třídy LR a obsahuje další dva ohraničující parametry. Prvním z nich je head H a druhým tail T. Tyhle dva parametry mohou být přidány z důvodu, abychom wrapperu zamezili prohledávání jinde, než se nacházejí data, která chceme extrahovat.

Parametr H nám označuje místo, kde končí hlavička dokumentu a tím omezíme vyhledávání pouze na vymezenou část dokumentu, která se nachází až za hlavičkou.

Druhý parametr T poukazuje na místo, kde je konec dokumentu. Respektive úsek dokumentu, který se má prohledávat až po parametr T. Pokud program tento parametr v dokumentu najde, ukončí se prohledávání.

Tento mechanismus je užitečný zejména pro dokumenty, kde jsou data uložena v tabulkách či podobných strukturách. Parametr L a R umožní odlišit jednotlivé řádky a pomocí H a T vymezíme oblast, kde se tyhle řádky nachází.



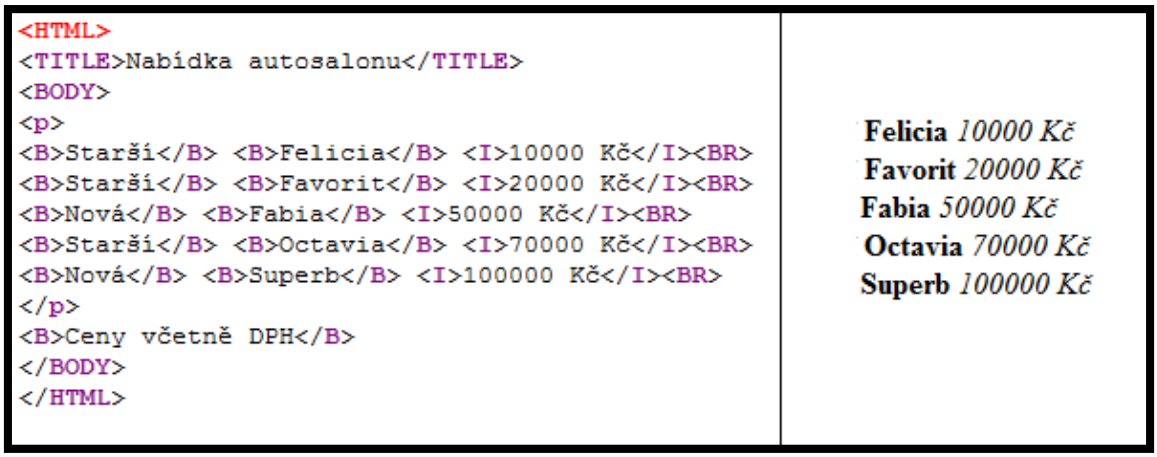
Obrázek 8 - Třída HLRT

Pokud bychom v tomhle případě použili pouze třídu LR a nastavili stejné parametry <B></B> a <I></I> tak by jako první text byl: „Dostupné automobily“ a k tomu cena „10000 Kč“. Přeskočilo by nám to text „Felicia“, což je samozřejmě špatně.

Přidáním parametru H a T dostane program přesně vymezenou oblast, kde prohledávat. Jako parametr H můžete zvolit <P> . Tím zajistíme, že wrapper začne extrahovat až data, která se nacházejí za tímto tagem. Jako parametr T zvolíme </P>. Tímto tedy označíme, kde má wrapper skončit s extrahováním a žádné další data nebudou prohledávána.

3.1.3 Třída OCLR

Třída OCLR (viz [**2**]) vychází také ze třídy LR, ale obsahuje další dva parametry. Opening a closing. Podobně jako u HLRT nám tyhle dva parametry označují oblast prohledávání, ale s tím rozdílem, že tyhle dva parametry jsou určeny pro každou dvojici parametrů LR. Wrapper tedy prohledává zdrojový kód, dokud nenalezne parametr O. Následně podle pravidel LR extrahuje text a hledá ukončovací parametr C. Tím extrahování textu pro jednu dvojici LR končí a hledá se další parametr O. Tato se operace se opakuje vždy se všemi čtyřmi parametry. Tím se liší od HLRT, jelikož tam je počáteční a koncový parametr určen pouze globálně pro celý dokument. Zamezíme tím extrahování nepodstatných dat mezi parametry LR v situaci, kdy máme pro jeden řádek více stejných tagů, jak bude vidět na Obrázek 9.



```
<HTML>
<TITLE>Nabídka autosalonu</TITLE>
<BODY>
<p>
<B>Starší</B> <B>Felicia</B> <I>10000 Kč</I><BR>
<B>Starší</B> <B>Favorit</B> <I>20000 Kč</I><BR>
<B>Nová</B> <B>Fabia</B> <I>50000 Kč</I><BR>
<B>Starší</B> <B>Octavia</B> <I>70000 Kč</I><BR>
<B>Nová</B> <B>Superb</B> <I>100000 Kč</I><BR>
</p>
<B>Ceny včetně DPH</B>
</BODY>
</HTML>
```

Felicia *10000 Kč*
Favorit *20000 Kč*
Fabia *50000 Kč*
Octavia *70000 Kč*
Superb *100000 Kč*

Obrázek 9 - Třída OCLR

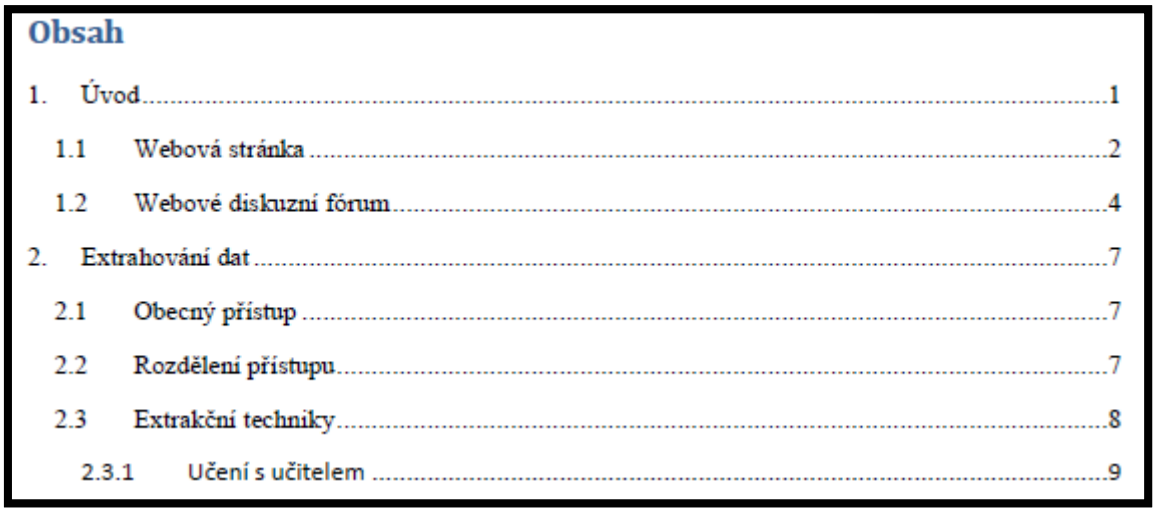
Na Obrázek 9 je vidět, že jeden záznam má dva stejné tagy, a to <B></B>. Program wrapper by tedy extrahoval i data, která pro nás nejsou v tuhle chvíli důležitá. Za pomocí parametru O, který nastavíme jako </B> zajistíme, aby se stav automobilu přeskočil a začalo se extrahovat až za tímto tagem. Ukončovací parametr C nastavíme jako <BR>. Máme zde tedy přesně vymezenou oblast pro každý záznam a wrapper bude extrahovat pouze data, která jsou pro nás důležitá.

3.1.4 Třída HOCLRT

Třída HOCLR, dle [**2**] je kombinací dvou předchozích, a to HLRT a OCLR. Využívá jak parametru H, který označí začátek prohledávané oblasti, respektive konec hlavičky a parametru T, který naopak určí, kde má prohledávání skončit. Každá dvojice parametrů LR je dále specifikována pomocí O a C parametrů.

3.1.5 Třída N-LR

Předchozí třídy wrapperů se využívaly hlavně ve spojení s tabulkami a extrakcí dat z nich. Třída N-LR, jak uvedl [**2**] je určena především pro zanořené (nestead) struktury, jako je např. obsah strukturované práce. Rozdíl je v rozdílných pravidlech pro každý záznam, jelikož ne všechny položky obsahu mají stejný počet zanořených položek.



Obrázek 10 - Třída N-LR obsah

Na Obrázek 10, který je použit z obsahu téhle diplomové práce pro ilustraci struktury zdrojových dat pro třídu wrapperu N-LR je naznačen rozdílný počet položek v jednotlivých sekcích. Bod „2.2 Rozdělení přístupu“ nemá již další položky na rozdíl od bodu „2.3 Extrakční techniky“, které obsahují další podsekci. Takové zanoření podsekcí by mohlo pokračovat a je tedy patrné, že nelze extrahovat data podle předchozích klasifikací tříd a je zapotřebí použití složitějšího algoritmu, který využívá rekurze a detekuje, zda se v uzlu nachází další objekty či nikoliv.

3.1.6 Třída N-HLRT

Třídu N-HLRT, dle [**2**] nebudu více rozebírat, ale pouze uvedu pro úplnost. Jedná se o téměř totožný postup, jako u předchozí třídy. Rozdílný je pouze základ třídy, který vychází ze HLRT, která je popsána výše. Použití taktéž pro zanořené struktury.

3.2 Stromové wrappery

Druhou skupinou wrapperů jsou stromové. Název vychází ze způsobu práce wrapperu se zdrojovým kódem. Rozdíl oproti řetězcovým je v tom, že zde se nepracuje s textem, ale se strukturou zdrojového kódu. Kód se může zpracovávat jako HTML struktura nebo je možné ji převést do XML. Obě varianty využívají stromové struktury, což znamená, že jednotlivé elementy kódu v sobě mohou obsahovat další elementy. Úkolem je tedy nalézt nadřazený element, ve kterém jsou obsažena data, která chceme extrahovat.

Dokument, který chceme zpracovávat, se načte do paměti, což nám umožňuje přistupovat k jednotlivým elementům náhodně a není nutné procházet vždy od začátku na konec. Což je rozdíl oproti řetězcovým wrapperům, které procházejí celý dokument sekvenčně. Je to výhodné pro webové stránky, které nejsou rozsáhlé, protože zde načtení celého zdrojového kódu do paměti může znamenat velice náročnou operaci.

3.3 Porovnání výsledků wrapperů

Je patrné, že každá třída řetězcových wrapperů má odlišné výsledky. Pokusím se tedy porovnat tři nejpoužívanější třídy. Zdrojový kód bude stejný pro všechny a budeme sledovat jak užitečné či neužitečné jsou jednotlivé výsledky.

```
<HTML>
<TITLE>Nabídka autosalonu</TITLE>
<BODY>
<B>Dostupné automobily</B>
<P>
<B>Starší</B> <B>Felicia</B> <I>10000 Kč</I><BR>
<B>Starší</B> <B>Favorit</B> <I>20000 Kč</I><BR>
<B>Nová</B> <B>Fabia</B> <I>50000 Kč</I><BR>
<B>Starší</B> <B>Octavia</B> <I>70000 Kč</I><BR>
<B>Nová</B> <B>Superb</B> <I>100000 Kč</I><BR>
</P>
<B>Ceny včetně DPH</B>
</BODY>
</HTML>
```

Dostupné automobily

Starší Felicia *10000 Kč*
Starší Favorit *20000 Kč*
Nová Fabia *50000 Kč*
Starší Octavia *70000 Kč*
Nová Superb *100000 Kč*

Ceny včetně DPH

Obrázek 11 - Zdrojový kód pro třídy wrapperu

Jedná se o jednoduchý HTML kód, na kterém lze ukázat rozdílné práce řetězcových wrapperů. Na následujícím obrázku budou zobrazeny výsledky všech tří wrapperů, kde budou patrné rozdíly v jejich zpracování. Nastavovací parametry zůstávají stejné, jako u popisu jednotlivých tříd. Na pravé straně Obrázek 11 je zobrazen vzhled webové stránky bez použití wrapperů.

LR třída	HLRT třída	OCLR třída
Dostupné automobily *10000 Kč* **Starší** *20000 Kč* **Nová** *50000 Kč* **Starší** *70000 Kč* **Nová** *100000 Kč* **Ceny včetně DPH**	**Starší** *10000 Kč* **Starší** *20000 Kč* **Nová** *50000 Kč* **Starší** *70000 Kč* **Nová** *100000 Kč*	**Felicia** *10000 Kč* **Favorit** *20000 Kč* **Fabia** *50000 Kč* **Octavia** *70000 Kč* **Superb** *100000 Kč*

Obrázek 12 - Porovnání tříd wrapperů

Z obrázku je tedy patrné, že nejlépe zde vyšla třída OCLR, jelikož zdrojový kód byl použit právě z téhle třídy a to z toho důvodu, že byl nejsložitější ze všech tří tříd.

První třída LR vzhledem k tomu, že bere vždy první výskyt tagů <B> </B> a <I> </I>, tak vyextrahovala text „Dostupné automobily“ a další dvojici, která je taktéž označena stejnými tagy přeskočila. Je to korektní chování, jelikož hledá výskyt tagů <I>. Po nalezení extrahoval cenu „10000 Kč“. Pro další postup opět bere první výskyt tagu <B> a následuje <I>. Vždy je tedy vynechána prostřední část a to konkrétně značka automobilu. K téhle informaci se wrapper nikdy nedostane.

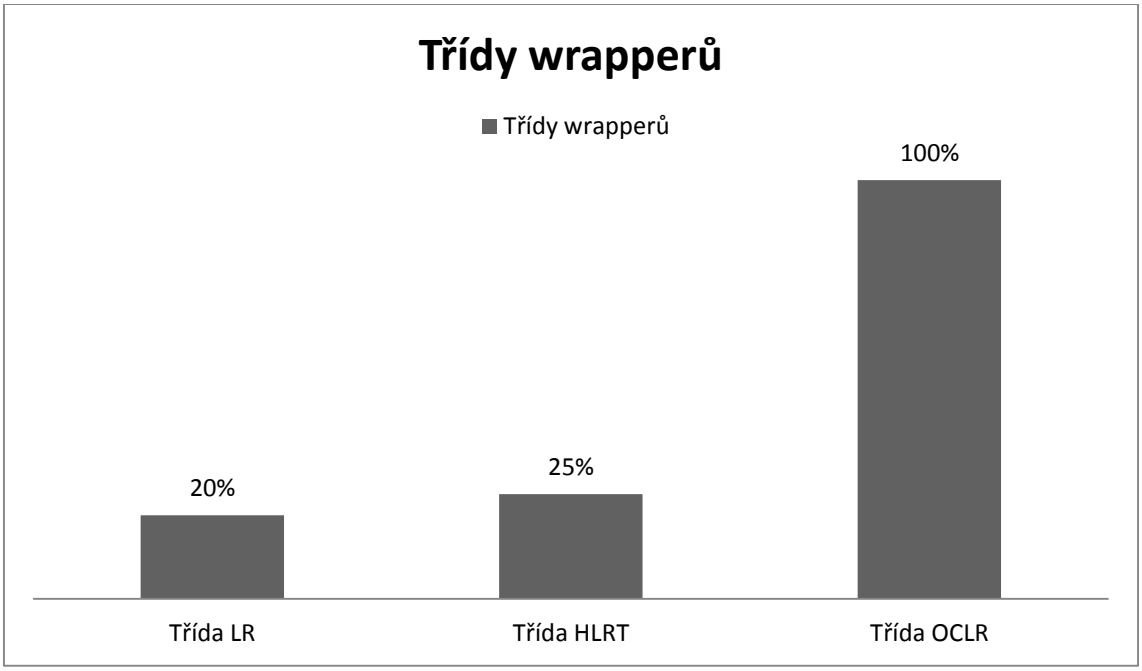
Druhá třída HLRT se již blíží ke správnému výsledku, ale vzhledem k tomu, že zde máme nastaveno pouze ohraničení na tento odstavec a jinak se postupuje stejně jako u předchozí třídy, je vynechána prostřední část, značka automobilu. Opět se bere jen první výskyt tagů <B> a <I>. V tomhle případě by stačila pouze malá změna nastavovacích parametrů, a výsledky by byly správné.

Poslední třída OCLR ukázala správné výsledky extrahování a to z toho důvodu, že je používána převážně pro data se stejnou či podobnou strukturou. Jako jsou tabulky či struktury typu seznam informací v posloupnosti za sebou. Pro každý řádek je nastaveno přesně dané pravidlo, které se v celém dokumentu stále opakuje.

Úspěšnost jednotlivých tříd wrapperů bychom mohli také ohodnotit pomocí procentuálního vyjádření. Za každou správně vyextrahovanou informaci by úspěšnost vzrostla. Cílem našeho extrahování jsou informace o značce vozidla a jeho cena. Jedná se o 10 položek, tudíž každá správná informace by znamenala přírůstek 10% a v případě vyextrahování špatné informace -5%. Podívejme se tedy na konkrétní výsledky, které korespondují s Obrázek 12.

Třída wrapperu	**Třída LR**		**Třída HLRT**		**Třída OCLR**	
Informace	Značka	Cena	Značka	Cena	Značka	Cena
Úspěšné	0%	50%	0%	50%	50%	50%
Neúspěšné	-30%	0%	-25%	0%	0%	0%
Celkem	20%		25%		100%	

Tabulka 1 - Porovnání výsledků tříd wrapperů



Třída OCLR tedy dopadla nejúspěšněji. Získala 100% dle zadání. U třídy LR je 0% za značku automobilu stejně jako u třídy HLRT, jelikož obě třídy vyextrahovaly první nález tagu opakujícího se vícekrát a druhý nález přeskočily. Posléze extrahovaly až cenu. Ta dopadla u všech tříd stejně dobře. Záporná hodnocení jsou udělena za informace, které jsme extrahovat nechtěli a i přesto se zobrazily mezi výsledky.

3.4 Nasazení wrapperů

Existují již systémy, kde jsou jednotlivé třídy wrapperů použity k extrahování dat. V téhle kapitole popíšu vybrané z nich včetně jejich principu fungování. Některé prvky funkčnosti těchto systémů budou využity i při mé implementaci.

3.4.1 WIEN

WIEN wrapper, (viz [**3**]) je velice uživatelsky přívětivý software, který pracuje přímo s prohlížečem uživatele. Tak jak je prohlížečem webová stránka zobrazena, uživatel vybere za pomocí označení textu, např. myší sekce ve kterých se nacházejí informace, které chce uživatel extrahovat. Wrapper posléze vyhodnotí vybrané sekce a sestaví pravidla pro vyextrahování vybraného textu. Využívá přitom třídy HLRT. Uživatel tedy pouze označuje data, která chce extrahovat a wrapper, pokud dokáže sestavit extrakční pravidla, která si posléze zapamatuje pro další extrakci, obstará vše ostatní.

3.4.2 STALKER

Systém STALKER, popsaný v [**4**] pracuje se strukturovanými a polostrukturovanými dokumenty. Využívá sadu trénovacích webových stránek k vytvoření extrakčních pravidel, které by dokázali pokrýt co nejvíce webových stránek. Seskupí tedy podobné modely, ke kterým jdou vytvořit stejná extrakční pravidla a ke zbylým která nemusejí mít žádné ekvivalentní webové stránky, vytvoří samostatná pravidla. STALKER pracuje se stromovou strukturou a koncové uzly, neboli listy stromu jsou právě ty místa, kde je potřeba nalézt pravidla pro správné extrahování. STALKER také pro každý koncový uzel udržuje cestu k rodičovskému uzlu, což umožňuje libovolný průchod celým stromem a extrahováním jednotlivých listů. Tento systém využívá pro svou práci ukládání jednotlivých uzlů k porovnávání, zda nová stránka je ekvivalentní naučenému pravidlu a pokud má list stromu stejnou cestu, jako předpis, jedná se o shodu a extrahování bude úspěšné.

4. Ukázka použití extrakčních pravidel

Obsah kapitoly

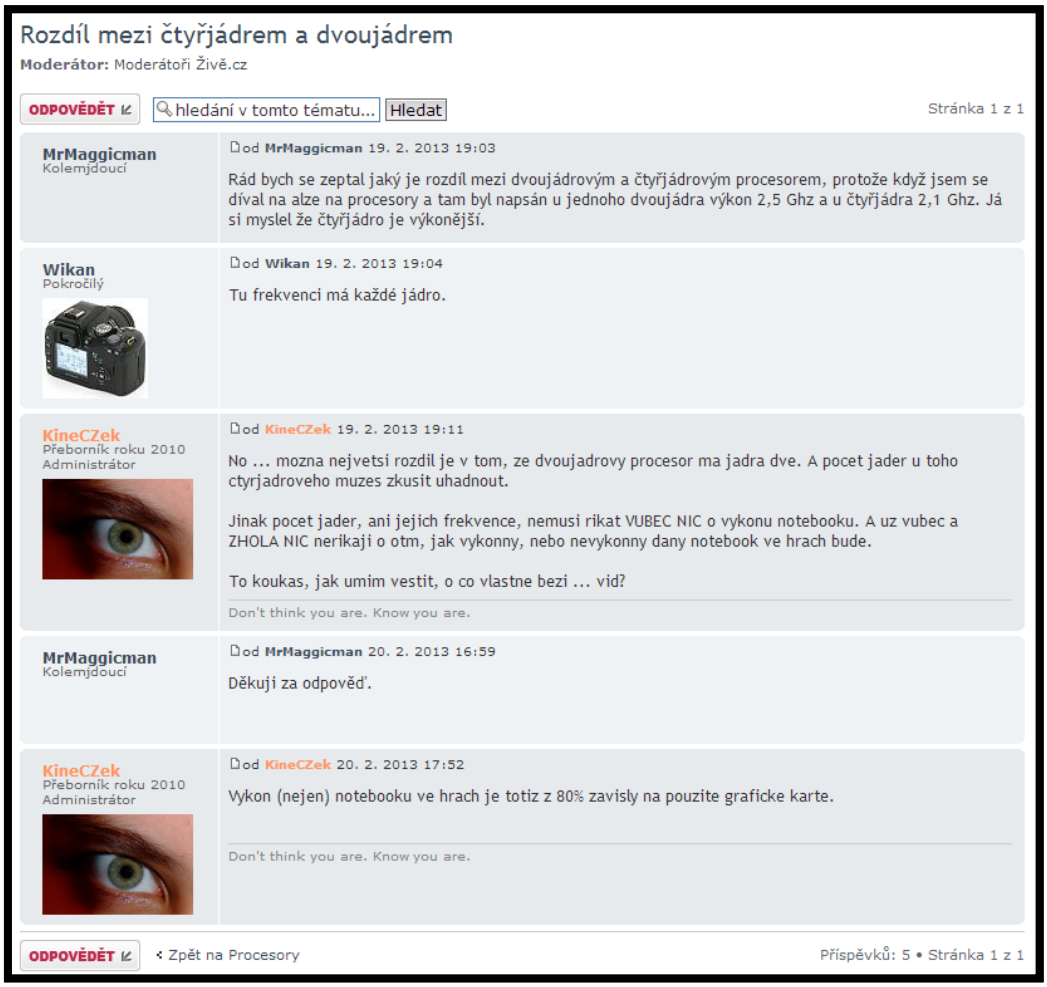
1) Vybrání webové stránky - diskuzního fóra
2) Hledání užitečných informací
3) Aplikace HLRT pravidel
4) Zhodnocení příkladu

Na základě výše uvedených technik uvedu příklad, na kterém předvedu, jak lze extrahovat data z webových diskuzních fór. Vybereme si jedno diskuzní fórum a v něm konkrétní příspěvek, ze které bychom chtěli vyextrahovat titulek o čem dané vlákno je, autora, datum publikování a samotný text ve kterém je daný problém popisován.

Vzhledem k tomu, že struktura příspěvků se opakuje, tak jednotlivé odpovědi autorovi mají stejnou šablonu a tudíž extrakce dat z odpovědí probíhá stejným způsobem jako u hlavního příspěvku, kvůli kterému bylo vlákno založeno.

4.1 Vybrání webové stránky – diskuzního fóra

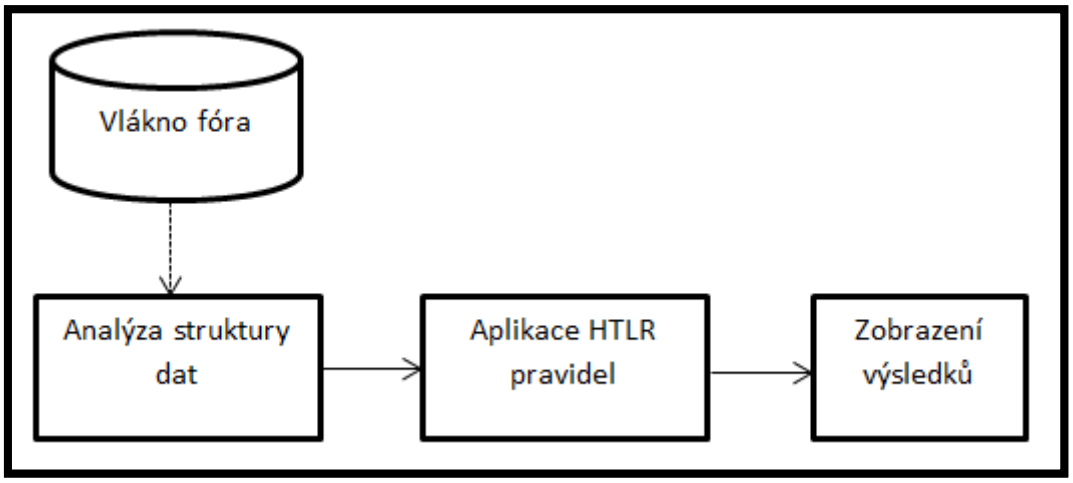
Jako příklad jsem zvolil fórum na serveru zive.cz a vybral jsem jedno téma, ze kterého bych chtěl získat pouze relevantní data. Samotné vlákno vypadá takto:



Obrázek 13 - Vlákno webového fóra

Na Obrázek 13 můžeme vidět vybrané vlákno, o kterém jistě víme, že se jedná o téma fóra a nemusíme zde testovat, zda to fórum je, či není. Na začátku je potřeba zvolit techniky, jak celý dokument, čili zdrojový kód budeme procházet a hledat pouze ty informace, které potřebujeme.

Jak jsem zmínil výše, budu se v téhle práci zabývat v praktické části výhradně poloautomatickým přístupem, čili tzv. „učení s učitelem“. Na předem známé množině stránek, které jsou s určitostí označené jako webové diskuzní fórum, je potřeba odhalit opakující se struktury, které v sobě nesou námi požadované informace. Použijeme řetězcových extrakčních technik a konkrétně metodu HLRT, jelikož díky svým nastavovacím parametrům můžeme dosáhnout požadovaných výsledků.



Obrázek 14 - Procesní schéma

Na Obrázek 14 máme naznačeno, jak probíhá celkové zpracování jedné webové stránky. V našem případě je vlákno webového diskuzního fóra jako zdroj dat. Data mohou být na stránce uložena staticky, ale ve většině případů slouží webová stránka pouze jako nástroj pro zobrazování dat z databáze. Při našem zpracovávání nezáleží, jakou formou jsou data na stránce získávána, protože pracujeme pouze s tím, co vidíme na obrazovce. Poté následuje analýza struktury dat. Za pomoci předem naučených pravidel z jiných webových fór detekujeme strukturu, která odpovídá některé pro nás známé. Na základě tohoto vyhodnocení, když už víme, jaké tagy by nás mohli zajímat, aplikujeme pravidla z techniky HLRT. Tímto získáme tížené data, která nás zajímají a můžeme je tedy zobrazit. Nemusí být v grafické formě, ale je důležité, abychom oddělily vyextrahovaná data od těch původních, což je našim cílem.

4.2 Hledání užitečných informací

Nyní, když známe na základě předem natrénovaných dat tagy či úryvky zdrojového kódu, kde by se mohli informace nacházet, začneme se samotným procházením a extrahováním dat.

První a určitě důležitá informace je titulek celého vlákna. Podle Obrázek 13 je to tedy „Rozdíl mezi čtyřjádrem a dvoujádrem“ a ve zdrojovém kódu ho nalezneme pod tagy „title“.

```
<title>Rozdíl mezi čtyřjádrem a dvoujádrem - Živě.cz</title>
```

Obrázek 15 – Titulek

Titulek dle Obrázek 15 nám oznamuje titulek celé webové stránky. Nadpis, který můžeme vidět na Obrázek 13, nám také zobrazuje nadpis příspěvku, ale funguje i jako odkaz na samotný příspěvek.

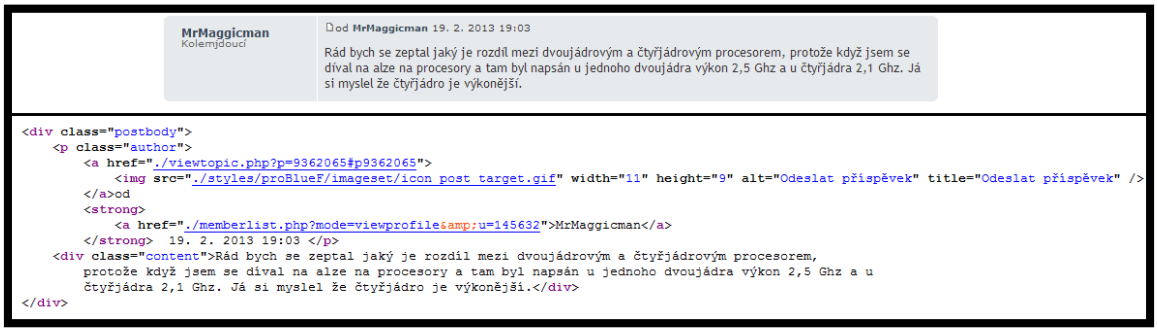
Další informace nám oznamuje jméno moderátora, který moderuje tohle vlákno a má tedy na starost, aby se zde dodržovaly pravidla příspěvku. Pro nás tedy informace, která není důležitá, bude přeskočena a nebude extrahována.

Následují ovládací prvky, jako tlačítko pro odpověď, vyhledávající políčko a tlačítko pro zahájení vyhledávání. Pro nás opět nedůležité.

Nyní následuje 5 stejných příspěvků, které obsahují informace, které nás zajímají nejvíce. Jsou graficky od sebe odděleny ohraničením a barevným označením. Jak již bylo zmíněno, jednotlivé příspěvky mají stejnou strukturu, takže pokud se nám podaří získat informace z prvního příspěvku, který byl založen autorem vlákna, ostatní postupy se budou opakovat.

4.3 Aplikace HLRT pravidel

Použitím metody HLRT máme tedy 4 různé parametry pro nastavení míst, kde chceme extrahovat. Jako parametr H zvolíme označení pro jednotlivý příspěvek. Parametr T nám ohraničí konec příspěvku tak, abychom nezasahovali v extrahování do jiných příspěvků. Pomocí množiny stránek, na kterých jsme prováděli tzv. „učení“ víme, že příspěvek je tedy v sekci označené počátečním tagem <div class> a ukončen tagem . Takovýchto sekcí je ale ve zdrojovém kódu mnoho. K tomu, abychom označili to správné místo, je potřeba ještě přesně určit jméno třídy neboli class. Pro nás je tedy důležitá sekce pojmenovaná jako „postbody“.



Obrázek 16 - Příspěvek

Nyní už máme vymezeno přesně to místo, kde se nachází námi požadované informace a máme tedy vyřešeny jednotlivé příspěvky. Zbývá ještě najít jednotlivé informace, které obsahuje každý příspěvek. K tomu nám poslouží označení L a R.

Pokud budeme tedy zdrojový kód procházet sekvenčně tak další informace která nás po titulku zajímá a je další v pořadí, je jméno autora, který vlákno založil a vložil první příspěvek. Pomocí parametru H a T jsme si vymezili příspěvek a nyní pomocí parametru L a R začneme procházet příspěvek. V našem případě tedy chceme, aby výstupem bylo jméno „MrMaggicman“.

Nacházíme se v sekci „postbody“ a autora nalezneme v podsekci, kde jako parametr L zvolíme a parametr R . Mezi těmito tagy se nalezneme autora i datum, kdy byl příspěvek vložen.

Po autorovi, další informace, kterou bychom chtěli získat je datum vložení příspěvku. V tomhle případě je datum součástí podsekce o autorovi, takže ho nelze extrahovat samostatně za pomocí jiných parametrů L a R. Jedinou možností jak tedy datum získat, je za pomocí vhodně nastaveného řetězcového parseru, který bude např. hlídat mezeru v textu a pokud následující slovo bude začínat číslicí, lze zbytek řetězce považovat za datum. To ovšem není v současné době předmětem prohledávání a extrahování informací ze zdrojového kódu, nýbrž práce s textem, který je již extrahován. V jiných strukturách html kódu nalezneme přímo tagy, ve kterých bude datum uloženo zvlášť a bude tak snadnější jej detekovat. V našem případě bychom ho museli tedy získat z řetězce v pozdější fázi.

Z příspěvku nám zbývá vyextrahovat už jen pouze samotný text, kvůli kterému celé vlákno vzniklo a který je nosičem hlavní informace. Na Obrázek 16 můžeme vidět, že sekce, ve které se text nachází je ohraničena parametry L a R, a to konkrétně L jako a R . Mezi těmito parametry nalezneme požadovaný text, který je poslední důležitou informací každého příspěvku.

Tímto postupem projdeme každý příspěvek vlákna a získáme tak kompletní přehled všech důležitých informací zproštěných od reklam, odkazů a jiných nežádoucích dat.

4.4 Zhodnocení příkladu

Jak již bylo řečeno, nelze zaručit, že za pomoci těchto HLRT parametrů lze extrahování aplikovat na všechny vlákna diskuzních webových fór různých webových serverů. Je třeba zvolit mechanizmus, který se bude starat o detekování podobných struktur a za pomoci předem natrénovaných dat obdobně procházet a extrahovat informace. Tento příklad nastavení wrapperu pro uvedený příklad bude samozřejmě fungovat na všechny vlákna nacházející na webovém portálu www.zive.cz . Po ruční analýze je třeba sestavit program, který bude extrakci dat provádět automatizovaně.

5. Návrh aplikace

5.1 Zadání

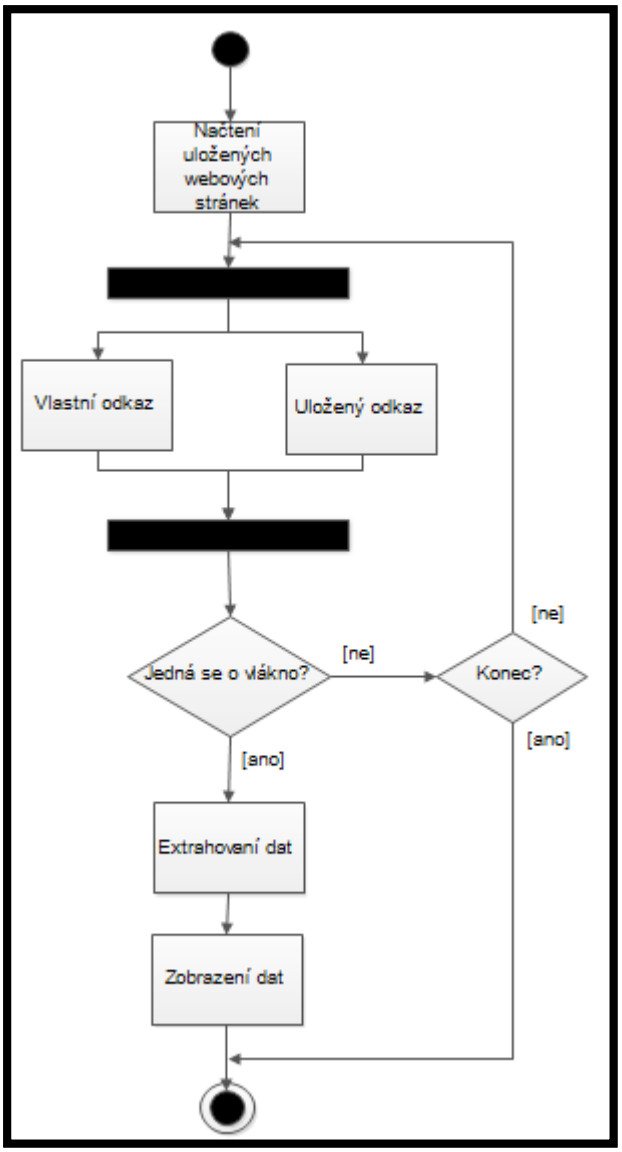
Cílem práce je vytvořit program, který bude extrahovat důležité informace z webových diskuzních fór. Vstupem bude odkaz na webovou stránku a na základě vyhodnocení, zda se jedná o diskuzní fórum či nikoliv bude na výstupu sada dat, které se v programu stanoví jako důležité.

Ostatní informace, jako reklamy a jiné data netýkající se daného příspěvku program přeskočí a nebude je extrahovat.

5.2 Funkční požadavky

1) Po spuštění aplikace budou předpřipraveny uložené webové stránky pro zobrazení správného chodu aplikace.
2) Uživatel bude mít možnost zadat své vlastní odkazy, které bude chtít prozkoumat a testovat.
3) Program bude analyzovat webovou stránku.
4) Informace úspěšně extrahované budou zobrazeny do needitovatelných políček.
5) Program zobrazí seznam autorů, kteří se zúčastnili tohoto příspěvku a podle vybraného autora se zobrazí i text jeho odpovědi.

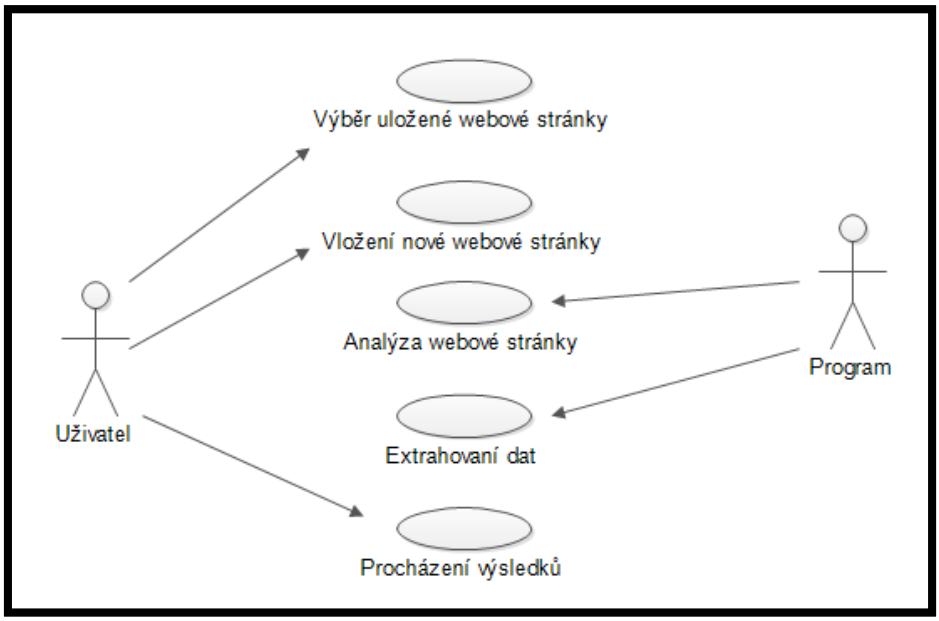
Ke snadnějšímu zobrazení chodu aplikace a jednotlivých aktivit slouží diagram aktivit na Obrázek 17, na kterém jsou patrné jednotlivé kroky, jak program pracuje.



Obrázek 17 - Diagram aktivit

.

Popis jednotlivých aktivit, které se v programu budou odehrávat, jsou zobrazeny na Obrázek 18 a to jak z pohledu uživatele, který bude program obsluhovat tak z pohledu softwaru samotného.



Obrázek 18 - Návrh aplikace - Use Case

Jak již bylo zmíněno, uživatel by měl mít možnost vybrat si buďto z uložených webových stránek anebo vložit vlastní odkaz. Další úkony již provede program, a to konkrétně analýzu webové stránky a popřípadě extrahovaní informací z ní. Na závěr bude mít možnost uživatel prohlédnout extrahovaná data včetně procházení odpovědí od jiných autorů.

Popisy jednotlivých případů užití jsou uvedeny v přílohách na konci dokumentu.

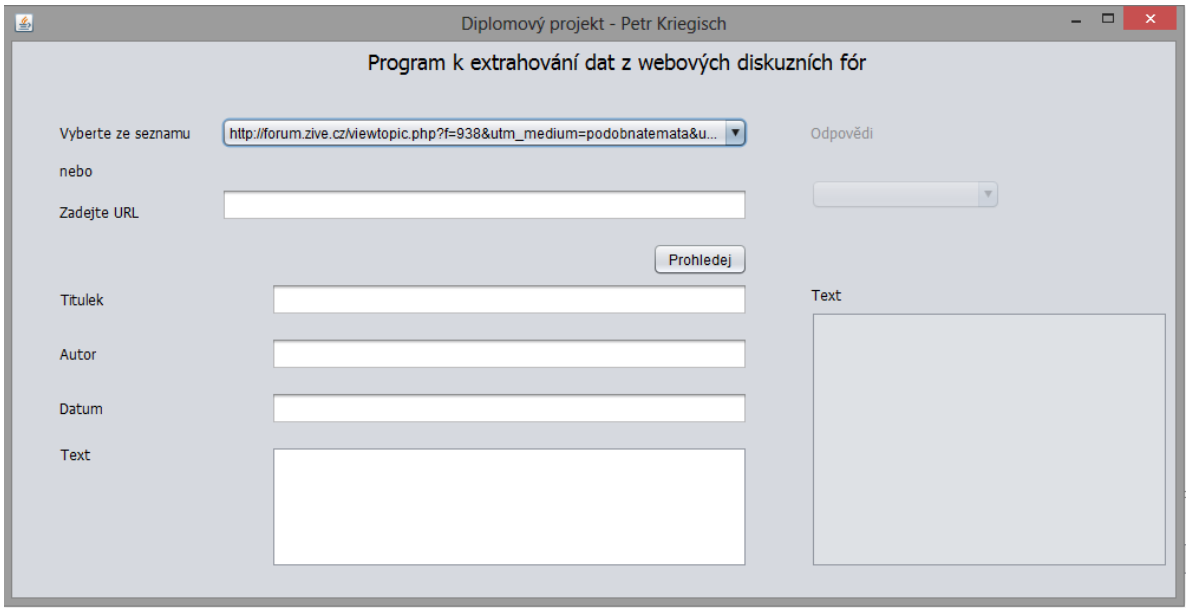
5.3 Technologické (nefunkční) požadavky

1) Vlastní GUI (grafické uživatelské rozhraní) - desktopová aplikace
2) Připojení k internetu pro připojení k webové stránce
3) Ověření webové stránky do 4 vteřin
4) Java ve verzi 7
5) Aplikace spustitelná přes ikonu
6) Wordlist součást aplikace

6. Implementace aplikace

Pro zpracování HTML dokumentu budu využívat HLRT techniku, která je popsána výše i na uvedeném příkladu. Podobnost lze pozorovat i se systémem STALKER. Program bude zpracovávat jakoukoli webovou stránku, a pokud bude zadána nekorektně, bude to uživateli oznámeno v podobě upozornění. K implementaci je potřeba znát také webové technologie, za pomocí nichž vytvořený program bude extrahovat informace. Ty nejdůležitější jsou uvedeny v terminologickém slovníku, který se nachází v sekci „Terminologický slovník“.

Podle návrhu aplikace jsem vytvořil desktopovou aplikaci, napsanou v jazyce Java. Ke správnému chodu aplikace je zapotřebí u spouštějícího souboru mít také wordlist obsahující seznam možných klíčových slov, obsahující autora. Více o tomto wordlistu bude zmíněno později v sekci o ověření webové stránky. Úvodní okno programu, vypadá následovně:



Obrázek 19 - Úvodní okno

Ke spuštění programu slouží javovský soubor, s příponou *.jar. Je tedy nutné mít na zařízení, na kterém se aplikace spouští nainstalovanou Javu alespoň ve verzi 7. Po spuštění programu se objeví následující okno, ve kterém si uživatel buďto vybere z předem uložených URL z webových diskuzních fór anebo zadá vlastní odkaz.

6.1 Ověření webové stránky

K tomu, abychom mohli webovou stránku začít zpracovávat jako vlákno jednoho z webových diskuzních fór, je nejprve nutné určit, zda se opravdu jedná o vlákno webového fóra, anebo je to

obyčejná webová stránka, u které nebudeme schopni informace extrahovat a ani nebude potřeba je extrahovat vzhledem k tomu, že se nebude jednat o vlákno webového diskuzního fóra. Doposud jsme tohle řešit nemuseli, jelikož uvedený příklad a popis technologií byl popisován pro jasně danou webovou stránku a strukturu, která byla opravdu korektní a obsahovala všechna kritéria, která webové diskuzní fóra obsahovat musí. Ovšem v našem programu, který bude na vstupu požadovat URL, bude tedy nejprve nutné ověřit, zda byla předložena správná webová stránka.

Abychom mohli ověření provádět, je potřeba stanovit si nějaká pravidla, která musí vlákno splňovat. Ve svém programu jsem si určil 3 různá pravidla, podle kterých se vyhodnocuje procentuální pravděpodobnost. Pokud pravděpodobnost přesáhne 50%, je stránka vyhodnocena jako webové diskuzní fórum a je tak dále zpracovávána. Pokud pravděpodobnost nedosáhne téhle hranice, stránka nebude považována jako vlákno webového diskuzního fóra a nebude dále zpracovávána. Tři pravidla, která jsem zvolil, tedy jsou:

- **Obsahující vhodné sekce** – jak již bylo popsáno výše, má implementace se zabývá poloautomatickým přístupem k extrakci. Z toho vyplývají naučená pravidla a techniky z předem projité množiny webových diskuzních fór. Takže pro ověření, zda i URL na vstupu zapadá do téhle množiny, jsem vytvořil soubor, který obsahuje pouze názvy tříd, které se nejčastěji objevují ve zdrojovém kódu takovéto webové stránky. Mezi názvy těchto tříd se zaměřuji pouze na klíčová slova týkající se autorů, jelikož každý příspěvek takový údaj musí obsahovat. Pokud program nalezne ve zdrojovém kódu 2 a více takovýchto tříd, pravděpodobnost, že se jedná o vlákno webového diskuzního fóra, se zvýší o 50%. Pokud obsahuje pouze jednu třídu s názvem týkajícího se autora, jedná se pouze o 30%, jelikož to může být pouze nějaký článek, který také autora může mít, ale nejedná se o fórum. Z toho je tedy patrné, že i když zdrojový kód obsahuje několik tříd s autory, neznamená to ještě, že se opravdu jedná o diskuzní fórum. Proto je důležité vzít v potaz i další podmínky.
- **Odkaz obsahuje klíčové slovo** – dalším kritériem, které se vyhodnocuje pro správné vyhodnocení webové stránky je, zda samotné URL obsahuje klíčové slovo „forum“ nebo „topic“. Ve většině odkazů, které byly v testovací množině, odkaz obsahoval jedno z těchto klíčových slov. Pokud tomu tak je u vstupního odkazu, pravděpodobnost, že se jedná o diskuzní fórum, se zvýší o 40%.
- **Ve zdrojovém kódu klíčové slovo –** poslední hodnotící podmínkou je, zda samotný zdrojový kód, který odpovídá webové stránce na vstupu, obsahuje odkaz na samo sebe. Téměř všechna webová fóra tento odkaz obsahují. Většinou to bývá právě titulek, který je nadpisem každého vlákna a nese v sobě odkaz na stejné vlákno. Pokud je takový odkaz nalezen, pravděpodobnost vzroste o 20%.

Kombinací těchto podmínek na závěr podle hodnoty, která udává pravděpodobnost, se zhodnotí, zda je to více než oněch 50% a jedná se tedy o vlákno webového diskuzního fóra či nikoliv. K tomu slouží okno, které obsahuje samotný výsledek a informuje nás o úspěšnosti. Vzhled tohoto okna je zobrazen na Obrázek 27, který se nachází mezi přílohami.

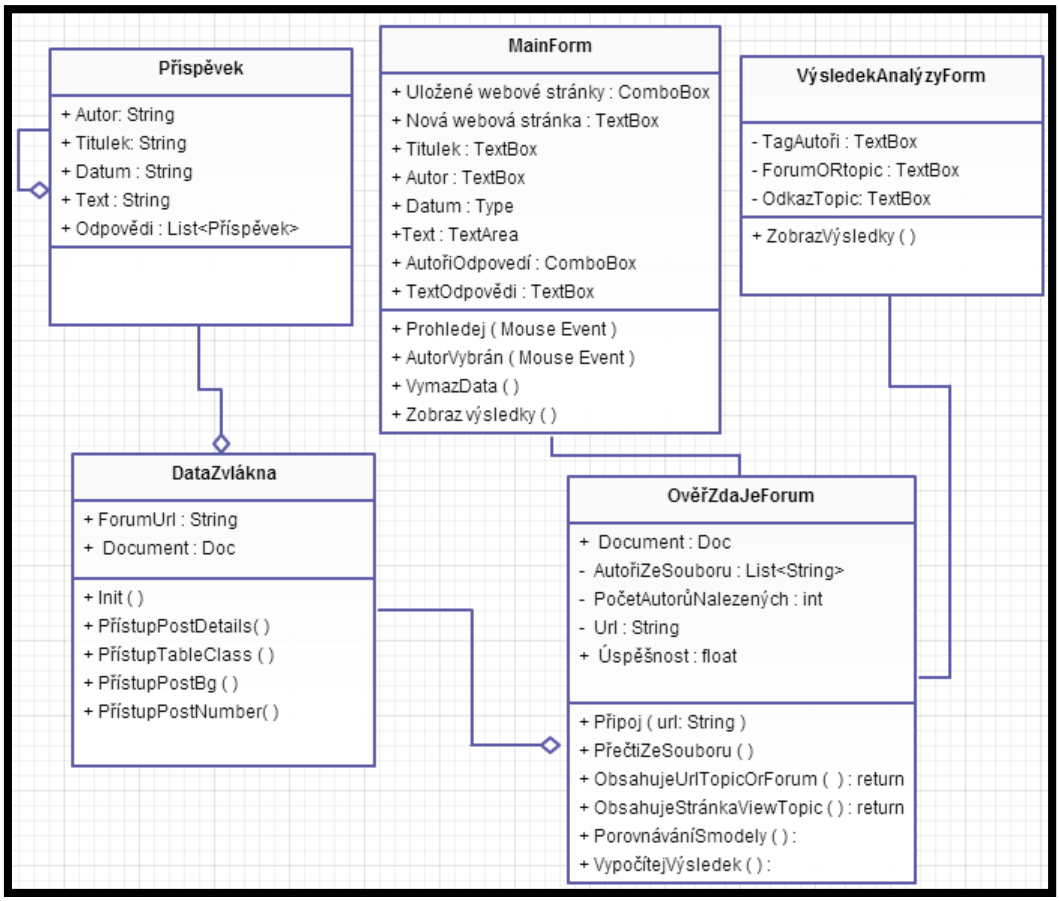
Postup programu lze vyjádřit podle pseudokódu, který je zobrazen na Obrázek 20.

```
VSTUP: webova_stranka;
list_klicovych_slov := nactiZeSouboru;
projdi => list_klicovych_slov
        if webova_stranka.class = klicove_slovo then
                pocet_autoru ++;
until => list_klicovych_slov.pozice = konec;
if pocet_autoru >= 2 then
        uspesnost+=50;
else if pocet_autoru > 0 then
        uspesnost+=30;
if webova_stranka.URL obsahuje "TOPIC" || "FORUM" then
        uspesnost += 40;
if webova_stranka.obsah obsahuje "viewtopic" then
        uspesnost+=20;
VYSTUP: if uspesnost > 50% then
                Zobraz: "Jedná se o vlákno diskuzního webového fóra";
         else
                Zobraz: "Nejedná se o vlákno diskuzního webového fóra";
```

Obrázek 20 - Ověření webové stránky, pseudokód

Tímto testem si tedy ověříme, zda bude prohledávání a popřípadě i extrahování dále pokračovat. Podobná funkčnost by mohla být využívána jistě i pro jiné účely, ať už pro vyhledávače, které by si udržovaly seznamy webových stránek, které jsou právě vlákny diskuzních fór anebo pro větší webový server, který by shromažďoval informace ze všech jemu známých webových fór a fungoval tak jako jedno veliké diskuzní fórum. To už ovšem není předmětem téhle diplomové práce.

Na Obrázek 21 je znázorněna iterace mezi třídami vytvořeného programu.



Obrázek 21 - Třídní diagram

„MainForm“ slouží jako hlavní okno aplikace, kde jsou zobrazovány výsledky extrahování. Třída, která obstarává ověřování, zda je webová stránka diskuzním fórem, se jmenuje „OvěřZdaJeForum“ a posléze třída „VýsledekAnalyzyForm“ zobrazí výsledky tohoto vyhodnocení.

Pokud je webová stránka diskuzním fórem, následuje extrahování dat a to konkrétně ve třídě „DataZvlákna“. Jednotlivé extrahované data se ukládají do objektu „Příspěvek“, který může obsahovat seznam odpovědí, které jsou uloženy do tohoto objektu jako seznam dalších objektů „Příspěvek“.

7. Experiment

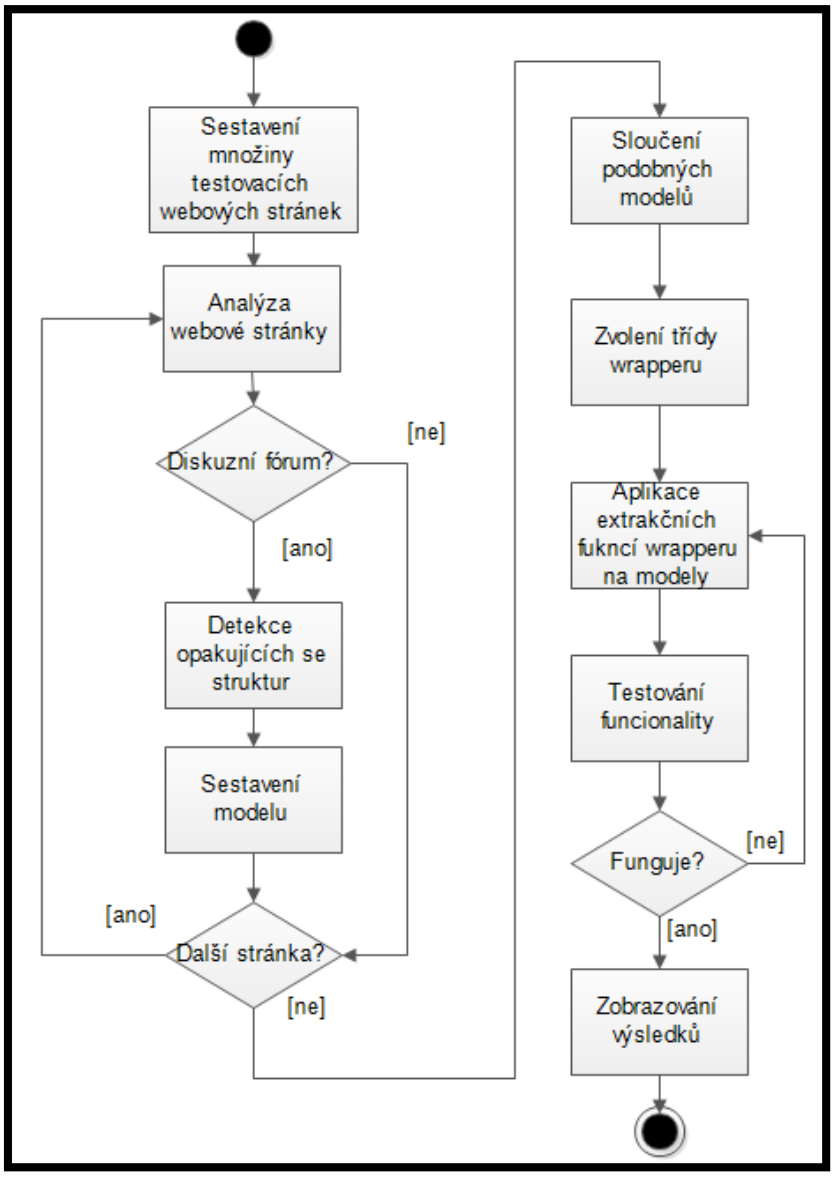
Obsah kapitoly

1) Přehled aktivit experimentu
2) Vytvoření testovací množiny webových stránek
3) Analýza jednotlivých webových stránek
4) Modely analýzy
5) Zvolení třídy wrapperu
6) Vytvoření programu a následné testování
7) Forma zobrazování informací a jejich další zpracování
8) Výsledky experimentu

7.1 Přehled aktivit experimentu

Nedílnou součástí dokumentu je také část, kde budu popisovat postupy, kterými jsem se během psaní programu zabýval a experimentoval. Bude to tedy souhrn všech aktivit, které vedly od nastudování trénovací množiny webových stránek až po dokončení programu, jehož cílem je extrahování dat z webových diskuzních fór.

Kompletní postup je zobrazen na Obrázek 22.



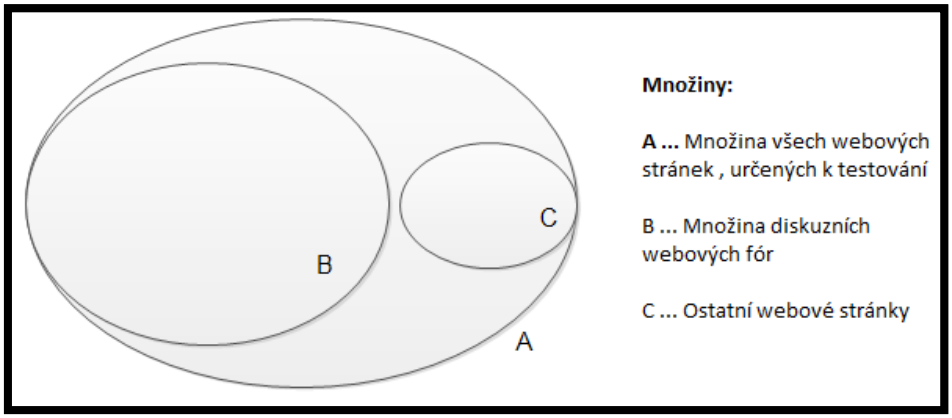
Obrázek 22 - Přehled aktivit postupu

Celý tento proces vede k výsledku, kdy uživatel na základě odkazu k jakékoliv webové stránce dostane informace o tom, zda je jedná o fórum či nikoliv. Pokud se opravdu jedná na základě vyhodnocení programu o vlákno diskuzního fóra, uživateli se zobrazí informace, které jsou v rámci daného odkazu důležité k extrahování. K tomu, abychom docílili takovéto funkcionality, je zapotřebí projít všemi aktivitami, zobrazenými na Obrázek 22.

7.2 Vytvoření testovací množiny webových stránek

V této fázi je potřeba vybrat sadu webových stránek, na kterých bude probíhat ruční analýza zdrojového kódu k rozpoznání, které webové stránky se týkají diskuzních fór a které jsou stránky nesoucí jiné data.

Celkem byla vytvořena množina [A] s 15 webovými stránkami. Pro můj postup jsem zvolil množinu 9 stránek [B], u kterých bylo známo, že se opravdu jedná o webová diskuzní fóra, a k tomu množinu [C] 4 webových stránek, které s diskuzními fóry nesouvisely a to z důvodu testování závěrečného extrahování, aby webové stránky z téhle množiny byly programem detekovány jako nevhodné pro extrahování. Zbylé 2 stránky byly vybrány z důvodu nejednoznačného určení, zda se jedná o množinu [A] nebo [B]. Z pohledu určení pravděpodobnosti tedy někde na hranici 50%.



Obrázek 23 - Množiny webových stránek

Z Obrázek 23 je patrné, že množina [A] a množina [B] jsou dvě rozdílné množiny a jsou od sebe striktně odděleny, aby se jednoznačně dalo určit, zda se jedná o pozitivní nález podmínek pro detekci fóra či nikoliv.

7.3 Analýza jednotlivých webových stránek

Pro každou webovou stránku z množiny [A] jsem provedl ruční analýzu k nalezení struktur či modelů, ve kterých se objevují jednotlivé příspěvky. Vzhledem k tomu, že všechny příspěvky v jednom konkrétním fóru mají naprosto identické struktury, bylo pro nalezení rozlišení příspěvků hledání opravdu opakující se sekcí tagů. Díky uspořádání zdrojového kódu nám práci usnadní nalezení hlavního uzlu, který obsahuje celý příspěvek a je pro všechny příspěvky na téhle webové stránce stejný. Postup pro nalezení a další zpracování je popsán v sekci: Ukázka použití extrakčních pravidel a nebudu ho tedy zde podrobněji popisovat.

7.4 Modely analýzy

Po dokončení analýzy všech stránek z množiny [B] mi vzniklo 9 modelů, které reprezentují určitou strukturu zdrojového kódu. Zobrazím jejich nejdůležitější uvození a rozlišení příspěvků v Tabulka 2.

Webová stránka (fórum)	Označení příspěvku
iPhone	<table class = „tablebg“>
Matematické	
Android	<div class = „post bg2“>
Travian	
4AllMobile	<table class = „tablebg“>
My-Mobile	<table class = „tablebg“>
Forum2Mobile	
TvFreak	
Živé	<div class = „post bg2“>

Tabulka 2 - Výsledek analýzy

Jak můžeme vidět v Tabulka 2, z 9 různých zdrojů nám vyšly 4 odlišná řešení. Což je přesně ten důvod, proč jsme prováděli podrobnější analýzu a objevili tak opakující se vzory na různých webových stránkách. Z toho máme 3 různé modely, které se opakují více než jednou a má tedy smysl je dále zkoumat a vytvářet extrakční funkce. Matematické fórum je mezi testovací množinou svou strukturou odlišné od všech ostatních. Lze také wrapper přizpůsobit téhle struktuře, ovšem bylo by to s vysokou pravděpodobností úspěšné pouze pro jeden konkrétní příklad, což už sklouzává spíše k manuálnímu přístupu extrahování dat, nikoliv poloautomatickému. Našim cílem je nalézt takové modely, které se dají využívat opakovaně pro více než jeden webový server. Tím jsme tedy dokončili další fázi, a to „sloučení podobných modelů“.

7.5 Zvolení třídy wrapperu

Následuje výběr třídy wrapperu, který je důležitý pro samotný přístup k extrakci dat. Zvolil jsem si třídu HLRT, jelikož pro účely, které budu potřebovat je ideálně stavěná a vzhledem k tomu, že vím, kde jednotlivé příspěvky začínají, je nastavovací parametr „H“ určen právě tímto řetězcem, který můžeme vidět ve druhém sloupečku Tabulka 2. Později tedy 3 modely struktury dat. Kdyby existoval nějaký pěvně stanovený standard, který by určoval, jakou strukturu má mít každé webové diskuzní fórum, byl by postup značně zjednodušen.

7.6 Vytvoření programu a následné testování

V další fázi musíme přepsat naše modely do programu samotného. Jak již bylo řečeno, já zvolil programovací jazyk Java a knihovnu JSoup. Jistě bychom mohli zvolit i jiný programovací jazyk, protože veškeré operace a funkcionalita není nijak svázána s programovacím jazyk, který jsem zvolil já.

Nyní po vytvoření programu následuje fáze, která se může opakovat i vícekrát, a to testování a upravování stávajících funkcí. I když máme přesně vymezena místa ve zdrojovém kódu, kde se nacházejí pro nás relevantní informace, které jsme se rozhodli extrahovat, nemáme ještě vyhráno. Z definice tříd wrapperu je patrné, že nám pomůže s nalezením a extrahováním informací mezi tagy, které jsme předem za pomocí extrahovacích funkcí vymezili, ovšem text, který se nachází uvnitř těchto tagů je již mimo oblast, kde by nám wrapper pomohl. Čili je na nás, abychom na základě testování a upravování stávajících funkcí doladili samotné extrahování. Může se např. stát, že mezi tagem, který nese jméno autora je jako text přidáno i datum, kdy byl příspěvek autorem publikován. Což bychom jistě chtěli jako informaci, která bude uvedena zvlášť. Je tedy potřeba zvážit, zda najdeme způsob jak parserovat univerzálně každý takový text anebo jej ponechat nezměněn.

Během testování oněch tří modelů, které vznikly sloučením podobných struktur dle Tabulka 2, jsem narazil při testování i jiných webových stránek, než se nacházejí v testovací množině na další model, který se opakoval ve více webových diskuzních fórech. Přidal jsem tedy i tento model mezi 3 již existující. Z toho důvodu je fáze testování nekonečnou a je potřeba zhodnotit samotnou úspěšnost modelů již vytvořených. Jistě bychom mohli nacházet nové a nové struktury, pro které by bylo potřeba vytvářet nové modely a pravidla.

7.7 Forma zobrazování informací a jejich další zpracování

Závěrečnou fází již je samotné zobrazení výsledků. Podle účelu, pro který má aplikace sloužit, zvolíme vhodný způsob, jak nakládat s extrahovanými informacemi. Pokud by měl program udržovat seznam navštívených webových stránek, stačilo by jistě data zapisovat do souboru ať už v podobě XML či prostého textu. Pokud bychom chtěli uchovávat i samotná data, která byla extrahována, bylo by nutné použít jako zdroj k uložení dat nějakou databázi. V mém případě se jedná o čisté zobrazení aktuálně vyextrahovaných dat, tudíž postačí zobrazení dat na obrazovce.

Není nutné data nijak dále ukládat a zpracovávat. Takže po vložení dalšího odkazu se všechny předchozí informace zahodí.

7.8 Výsledek experimentu

Po dokončení celého procesu vývoje programu, včetně testování na předem vybrané množině webových stránek nastává čas k ověření funkčnosti, úspěšnosti detekce a následné extrakci informací z webových stránek, které nejsou nijak spojené s testovací množinou a které by uživatel používající program začal využívat. Zde se tedy projeví univerzálnost našich wrappovacích modelů, podle kterých program pracuje.

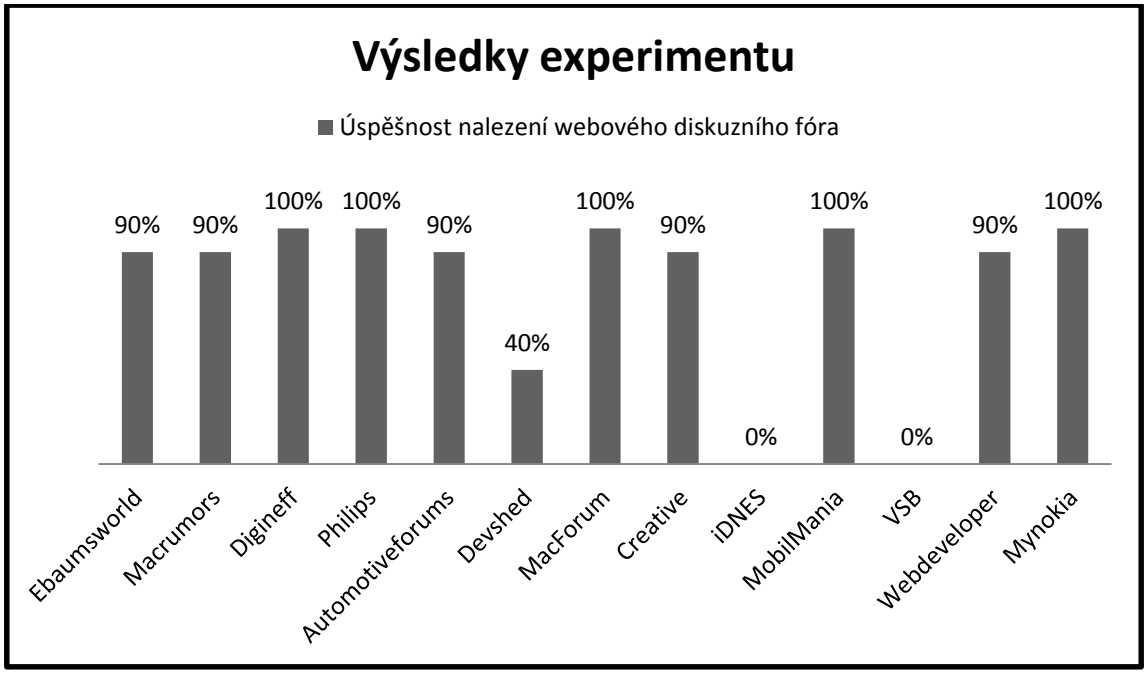
Opět tedy vybereme sadu webových stránek, které budou z jiných webových serverů, než se nacházely testovací webové stránky a budeme analyzovat, na kolik náš program pracuje správně.

Webová stránka (fórum)	Úspěšnost	Označení příspěvku
Ebaumsworld	90%	
Macrumors	90%	<table class = „tborder“>
Digineff	100%	<div class = „post bg2“>
Philips	100%	
Automotiveforums	90%	<table class = „tborder“>
Devshed	40%	Neznámo
MacForum	100%	<div class = „post bg2“>
Creative	90%	
iDNES	0%	Není vlákno
MobilMania	100%	<div class = „post bg2“>
VSB	0%	Není vlákno
Webdeveloper	90%	Neznámo
Mynokia	100%	<table class = „tablebg“>

Tabulka 3 - Výsledky experimentů

Tabulka 3 nám zobrazuje přehled náhodně vybraných webových stránek, ovšem spíše zaměřených na webová diskuzní fóra, abychom si ověřili, jak velké množství webových stránek máme pokryto pomocí vytvořených modelů pro extrahování. Mezi webovými stránkami jsou i 2, které se netýkají diskuzního fóra a tudíž jsou vyhodnoceny jako úspěšnost 0%, což je v pořádku a odpovídá to zadání. V takovémto případě program vyhodnotí webovou stránku správně, že se opravdu nejedná o vlákno webového diskuzního fóra.

Ze 13 webových stránek bylo celkem 11, které se dají považovat za webové diskuzní fórum. Z tohoto počtu jsme u 9 webových stránek dokázali i vyextrahovat úspěšně informace, které se na webové stránce nacházely a byly cílem našeho extrahování. Splňovali tedy jeden ze čtyř modelů struktury dat, které máme v programu definovány. I když u zbylých dvou webových stránek program správně detekoval, že se opravdu jedná o webové diskuzní fórum, nedokázali jsme vyextrahovat informace. Je to dáno neznámou strukturou zdrojového kódu, ve kterém jsou informace uloženy a bylo by tak potřeba vytvořit nový model.



Nulová úspěšnosti nastala právě ve dvou případech, kde se opravdu o vlákno webového diskuzního fóra nejednalo. V jednom případě došlo k detekování na úrovni 40%, ale webová stránka nebyla extrahována ani považována za webové diskuzní fórum, i když je jednalo o diskuzní fórum. Čili nesprávná detekce způsobená neznámým modelem struktury zdrojového kódy a tím souvisejícím vyhodnocením webové stránky.

7.8.1 Metoda pozitivní predikce

Výsledky experimentu lze zhodnotit podle metody negativní predikce, kde základem je kontingenční tabulka 2x2 vyjadřující 4 různé situace, které mohou při vyhodnocování nastat.

Celkový počet webových stránek = 13

	Pozitivní	**Negativní**	**Popis**
Pozitivní test	**TP (true positive)** Skutečně pozitivní 10	**FP (false pozitiv)** Falešně pozitivní 0	prediktivní hodnota pozitivního testu PPV = TP/(TP+FP) PPV =10/(10+0)= **100%**
Negativní test	**FN (false negativ)** Falešně negativní 1	**TN (true negativ)** Skutečně negativní 2	prediktivní hodnota negativního testu NPV= TN/(FN+TN) NPV= 2/(1+2)**≈ 66,67%**
	Senzitivita = TP / (TP + FN) = 10 / (10 + 1) ≈ **90,1%**	Specificita = TN / (FP + TN) = 2 / (0 + 2) = **100 %**	

PPV – Prediktivní hodnota pozitivní testu, jindy také uváděna jako hodnota „Precision“ nám ukazuje pravděpodobnost, že jev je skutečně pozitivní, když test vyšel pozitivně. 100 % poukazuje na přesnou detekci.

NPV - Prediktivní hodnota negativního testu nám ukazuje pravděpodobnost, že jev je skutečně negativní, když test vyšel negativně. Zde pravděpodobnost 66,67 % poukazuje na horší přesnost. Je to dáno i menším počtem vzorků, jelikož jeden FN výrazně pravděpodobnost srazil.

Senzitivita – Celková míra skutečně pozitivních nálezů nebo také hodnota „Recall“. Pravděpodobnost, že program korektně ohodnotí webovou stránku je 90,1%.

Specificita – Míra skutečně negativních nálezů. V našem případě 100%, takže negativní nálezy byly úspěšně detekovány.

7.8.2 F-skóre

F-Skóre vyjadřuje přesnost a úplnost harmonického průměru. Je založeno na dvou hodnotách, ze kterých se dopočítává výsledné skóre. K tomuto výpočtu použijeme hodnoty z předešlého testování, a to konkrétně Precision a Recall.

Výpočet F-skóre je:

$$\mathrm{F} = 2 \,.\, \frac{\mathrm{precision}\,.\ \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

Po dosazení vypočtených hodnot Precision a Recall dostaneme:

$$\mathrm{F} = 2 \,.\, \frac{1\,.\,0{,}91}{1 + 0{,}91} = 0{,}953 = \mathbf{95{,}3}\%$$

Vzhledem k tomu, že nám F-skóre vyšlo blízké k 1 čili 100%, přesnost je velice vysoká a rozhodně se nejedná o nahodilé vyhodnocování jevu ohodnocení webové stránky.

7.8.3 Metthewsův korelační koeficient

Metthewsův korelační koeficient je využíván u strojového učení jako měřítko vyhodnocování. Výsledná hodnota je v rozmezí -1 až +1. Hodnota rovna +1 značí perfektní predikci výsledku, hodnota rovná 0 je rovna náhodné predikci a -1 značí naprosto špatnou a nic neříkající predikci vyhodnocovaných objektů. Opět k výpočtu využijeme hodnot vypočtených v předešlých technikách, a to konkrétně „Metoda pozitivní predikce“ a hodnoty TP, FP, TN a FN.

Výpočet korelačního koeficientu je dán vzorcem:

$$\mathrm{MCC} = \frac{\mathrm{TP}.\mathrm{TN}\ -\ \mathrm{FP}\,.\ \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP}).\,(TP + FN).\,(TN + FP).\,(\mathrm{TN} + \mathrm{FN})}}$$

Po dosazení vypočtených hodnot dostaneme:

$$\mathrm{MCC} = \frac{10.2\ -\ 0\,.1}{\sqrt{(10 + 0).\,(10 + 1).\,(2 + 0).\,(2 + 1)}} = \mathbf{0{,}78}$$

V našem případě vyšel korelační koeficient roven 0,78, což se dá označit za velice slušnou predikci.

8. Závěr

Během studování struktury webových diskuzních fór a jejich odlišností od webových stránek, které nenesly prvky diskuzních fór, jsem narazil na mnohé rozdíly, které vedly k úspěšnému odhalení potřebných opakujících se modelů struktury zdrojového kódu pro sestavení extrakčních pravidel pro wrapper.

I když jsem ve své práci začal pracovat se třemi takovými modely, během testování funkčnosti jsem narazil na jeden další, který se doimplementoval ke zvýšení efektivnosti práce wrapperu. Samozřejmě při testování dalších webových stránek by se objevily nové modely a to dává prostor k dalšímu rozšiřování aplikace. Součástí je také wordlist nejčastěji se opakujících klíčových slov označujících v příspěvku autora, který se dá snadno rozšiřovat.

Do budoucna by bylo jistě užitečné aplikaci rozšířit o zdroj ukládaných dat, kam by bylo možné uložit extrahované informace. Lze by se tak snadno dala udržovat historie navštívených webových stránek a možnost práce v offline režimu.

Vytvořený program dosáhl na základě zhodnocení výsledků experimentů velice dobré úspěšnosti i na náhodně vybíraných webových stránek. Je velice pravděpodobné, že nelze sestavit program, který by dosáhl 100% úspěšnosti z hlediska extrahování informací. Pro detekci, zda se u předloženého odkazu jedná o diskuzní fórum, je úspěšnost odhalení vyšší.

Mnou zvolená technika implementace funkčnosti programu využila metodu třídy wrapperu HLRT, která sice ve výše uvedeném porovnání nevyšla jako nejlepší, ovšem pro extrahování jednotlivých příspěvků je velice efektivní. Tím se také program podobá systému STALKER pracující velice podobným způsobem.

Terminologický slovník

Xpath - Jazyk XPath slouží k identifikaci objektů neboli uzlů v objektovém modelu, nejčastěji v XML, dle [**5**]. Jeho hlavní rysem je vyhledávání relativních cest k uzlům či atributům tak, aby vrátil množinu výsledků. Výsledem může být jeden výraz nebo množina výrazů, která odpovídají vstupní podmínce. Pokud jazyk XPath nenalezne žádný výraz odpovídající vstupní podmínce, vrátí prázdnou množinu. Lze zde vidět určitou podobu s jazykem SQL, kde výsledkem je také množina výrazů, jeden výraz či žádný.

V našem případě se vlastnosti jazyka XPath hodí při prohledávání zdrojového kódu a vyhledávání tak uzlů, které obsahují další relevantní elementy a pro nás důležitá data. Vše bude zpracováváno za pomocí knihovny JSoup, kterou představím později a která umí pracovat zdrojovým kódem a vyhledávat v něm.

Java - Java je objektově orientovaný programovací jazyk a patří mezi nejpoužívanější programovací jazyky na světě, (viz [**6**]). Řadí se mezi multiplatformní, což umožňuje spustit software vytvořený v jazyce Java na mnoha zařízeních nezávisle na operačním systému. Aplikace mohou být provozovány na mobilních zařízeních, desktopových počítačích, systémech pracujících s čipovými kartami či distribuovaných systémech pracujících na propojených počítačích po celém světě.

JSoup - JSoup je Javovská knihovna, popsaná v [**7**], sloužící pro zpracování webových stránek a pro práci s nimi. Základem je tedy procházení HTML kódu, možnost zpracování pomocí DOM, CSS a jquery. Nové verze umí pracovat i s HTML5 specifikací. JSoup dokáže zpracovávat tři hlavní typy vstupu:

- **Řetězec** – nejjednodušším způsobem, jak knihovně předat vstup, který chceme parserovat a dále jej zpracovávat je string, neboli řetězec, který obsahuje kus HTML kódu.
- **Soubor** – druhým způsobem, jak zpracovat úryvek či celý dokument HTML je varianta se souborem na vstupu, kde je uložen zdrojový kód. Nejčastější typ souboru je textový soubor *.txt. Výhodou tedy je, že se nemusíme starat o otevírání čí zavírání souboru jako takového. O vše se postará knihovna JSoup, které stačí zadat pouze cestu k soboru.
- **URL** – asi nejčastějším použitím knihovny je v kombinaci se samotným odkazem na webovou stránku. Zde stačí zadat pouze URL odkazující se na jednu konkrétní

stránku, kterou chceme zpracovávat, a kompletní zdrojový kód se načte do paměti. Zde se využívá DOM.

HTML – HyperTextMarkupLanguage je jazyk, který slouží pro značkování webových stránek a pro jejich tvorbu, (viz [**8**]). Tak jako má každý jazyk, například čeština svá slova, stejně tak HTML má své slova, značky, které se nazývají tagy a pomocí nichž se utváří samotný vzhled a struktura webové stránky.

Tagy máme párové i nepárové. To znamená, že ne každý text musí být ohraničen z leva i zprava tagem. Mezi párové patří např. a , který uvozuje místo v dokumentu, které se označuje jako odstavec. Nepárovým tagem je např. <hr>, který do dokumentu umístí horizontální čáru, čili oddělení dvou objektů na stránce od sebe.

Tagy se píšou malými písmeny, ale není to nutnost, spíše doporučení vhledem k tomu, že u XHTML, které je rozšířením klasického HTML se již musí psát malými písmeny.

Každá webová stránka má svůj vzhled a strukturu definovanou podle uspořádání jednotlivých tagů, což odlišuje jednotlivé webové stránky od sebe. Ale je také potřeba dodržet základní strukturu, která je pro všechny webové stránky stejná.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
  </head>

  <body>
  </body>

</html>
```

Obrázek 24 - Základní struktura HTML kódu

Na Obrázek 24 můžeme vidět, jaké tagy jsou společné pro všechny webové stránky.

První řádek nám říká, jaká verze HTML byla použita. Dále již tento údaj není na samotné webové stránce uveden. Následuje samotné označení HTML dokumentu, mezi kterým se nachází všechny tagy a data webové stránky. Následuje hlavička (head), kde se nachází metadata o celé webové stránce. Např. název, kódování, jazyk, popis a další. Mezi tagy<body> a </body> se nachází již samotná data a další tagy, které utváří vzhled webové stránky.

CSS – Kaskádové styly - Kaskádové styly, dle [**9**] slouží pro popis různého zobrazování webových stránek napsaných v jazyce HTML, XHTML nebo XML. Byl navržen organizací W3C jako standard a v současné době existují dvě specifikace CSS1,CSS2 a pracuje se na verzi CSS3. Hlavní myšlenkou je oddělit vzhled webové stránky od jejího obsahu a struktury. Samotné HTML

obsahuje možnosti, jak měnit vzhled webové stránky a kaskádové styly tak nejsou nutností při vytváření webové stránky.

Kaskádové styly mohou být vytvářeny a definovány přímo ve zdrojovém kódu HTML, ovšem nejčastěji se využívá vytvoření externího souboru, na který se ve zdrojovém kódu vytvoří odkaz.

Každý definovaný styl se skládá ze dvou částí. První je tzv. selektor, který definuje, kterého úseku v kódu HTML se bude styl týkat. Může tím být celý dokument nebo pouze jeho úryvek. Druhou části každého kaskádového stylu je blok deklarací, ve kterém definujeme vlastnosti, které chceme nastavit. Za každou vlastností následuje dvojtečka a hodnota vlastnosti, na kterou ji chceme nastavit. Jednotlivé vlastnosti jsou od sebe odděleny středníkem.

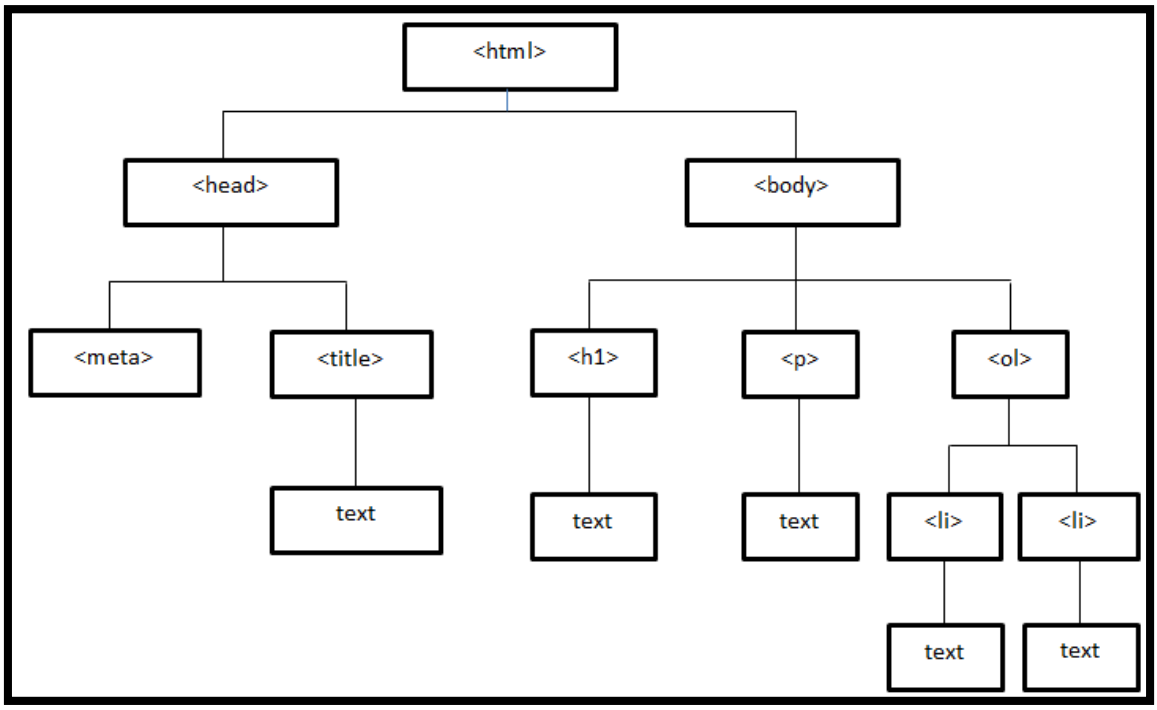
```
p {
  color: blue;
  font-weight: bold
  }
```

Obrázek 25 - Použití CSS v externím souboru

Na 20 je použit styl, který způsobí, že všechny odstavce <p> budou mít modrou barvu písma a zároveň písmo bude tučně. Obrázek pochází z použití v externím souboru. V samotném HTML kódu vytvoříme odkaz na soubor, který je pojmenován „styl.css“ pomocí tagu<link rel=”stylesheet” type=”text/css” href=”styl.css”>.

Pokud bychom chtěli stejný styl použít přímo v kódu HTML, je potřeba kód stylu ohraničit tagy<style></style>.

DOM - Jedná se o objektově orientovaný model (Document object model), reprezentující HTML nebo XML strukturu, popsaný v [**10**]. DOM je platformě nezávislé řešení a přináší také nezávislost na programovém zpracování. Díky tomu, že se jedná o objektově orientovanou techniku, vyplývá z toho výhoda stromového uspořádání objektů, což koresponduje se zmíněnými XML a HTML. Další vlastností přinášející technologie DOM je načtení zpracovávaného dokumentu do paměti umožňující tak náhodnému přístupu k jednotlivým objektům a přecházením mezi nimi. Pokud bychom neměli celý dokument načtený do paměti, muselo by se procházet při každém přístupu k části kódu v HTML či XML od začátku. Což je z hlediska složitosti jistě náročnější. Oproti tomu je potřeba dávat pozor na objemnost celého dokumentu, aby se nestalo, že načtený dokument bude velice rozsáhlý a způsobí tak zahlcení paměti. Pokud bychom narazili na takový dokument, je vhodné použití SAX model. Neboli sekvenční přístup. Jak takový model dokumentu vypadá, můžeme vidět na Obrázek 26.



Obrázek 26 – DOM

Seznam obrázků

Literatura

1. LIU, B. In: *http://cs.famaf.unc.edu.ar/~laura/llibres/wm.pdf.gz* [online]. 2007.

2. ROWE, M. *Wrapper Implementation for Information Extraction from House Music Web Sources*. 2005.

3. KUSHMERICK, N. Wrapper Induction for Information Extraction. 1997.

4. ION MUSLEA, S. M. C. K. STALKER: Learning Extraction Rules for Semistructured, Web-based Information Sources. 1998.

5. w3schools [online]. 2013. Dostupné také z: http://www.w3schools.com/xpath/xpath_intro.asp

6. LEAHY, P. Java About. In: *Java* [online]. [cit. 2013. Dostupné z: http://java.about.com/

7. JSoup. *JSoup* [online]. [cit. 2013-duben]. Dostupné z: http://jsoup.org/

8. *Pestujemeweb* [online]. [cit. 2013]. Dostupné z: http://www.pestujemeweb.cz/obsah/html/html-tagy-struktura.php

9. WIKIPEDIE. In: *Kaskádové styly* [online]. 2013. Dostupné také z: http://cs.wikipedia.org/wiki/Kask%C3%A1dov%C3%A9_styly

10. W3. *W3/DOM* [online]. [cit. 2013-duben]. Dostupné z: http://www.w3.org/DOM/

Seznam příloh

Příloha I. - Případy užití

1) Výběr uložené webové stránky

Aktér: Uživatel

Rozsah: Desktopová aplikace pro extrahování dat

Úroveň: Uživatelská

Úspěch: Zobrazení vyhodnocení webové stránky

Spouštěč: Spustitelný soubor s koncovkou *.jar

Hlavní scénář:

1. Program zobrazí seznam uložených odkazů
2. Uživatel zvolí jeden z dostupných odkazů
3. Uživatel klikne na tlačítko pro zpracování webové stránky
4. Zobrazení statistiky webové stránky

Alternativa: Chyba komunikace

V kroku 4 program detekuji chybu v komunikaci a oznámí tuto skutečnost uživateli, který má možnost opakovat krok 3 nebo vybrat nový odkaz.

2) Vložení nové webové stránky

Aktér: Uživatel

Rozsah: Desktopová aplikace pro extrahování dat

Úroveň: Uživatelská

Úspěch: Zobrazení vyhodnocení webové stránky

Spouštěč: Spustitelný soubor s koncovkou *.jar

Hlavní scénář:

1. Program zobrazí textové pole pro vložení odkazu webové stránky
2. Uživatel vloží odkaz
3. Uživatel klikne na tlačítko pro zpracování webové stránky
4. Zobrazení statistiky webové stránky

Alternativa: Chyba komunikace

V kroku 4 program detekuji chybu v komunikaci a oznámí tuto skutečnost uživateli, který má možnost opakovat krok 3 nebo vybrat vložit odkaz.

3) Analýza webové stránky

Aktér: Program

Rozsah: Desktopová aplikace pro extrahování dat

Úroveň: Programová

Úspěch: Zobrazení vyhodnocení webové stránky

Hlavní scénář:

1. Program detekuje, zda je vložen nový odkaz či vybrán jeden z uložených
2. Program zpracuje předložený odkaz na základě uživatelova požadavku
3. Program zobrazí statistiky webové stránky
4. Při detekci webového diskuzního fóra pokračuje v extrahování dat

Alternativa: Chyba komunikace

V kroku 2 program detekuji chybu v komunikaci a oznámí tuto skutečnost uživateli, který má možnost vybrat jinou webovou stránku.

Alternativa: Není webové diskuzní fórum

V kroku 3 program nenalezne splňující podmínky pro úspěšné vyhodnocení webové stránky jako diskuzní fórum a oznámí tuto skutečnost uživateli, který má možnost vybrat jinou webovou stránku.

4) Extrahování dat

Aktér: Program

Rozsah: Desktopová aplikace pro extrahování dat

Úroveň: Programová

Úspěch: Zobrazení dat o autorech a jejich odpovědích

Hlavní scénář:

1. Po detekci webového diskuzního fóra program analyzuje strukturu webové stránky
2. Program porovnává strukturu webové stránky se známými modely
3. Extrahování dat dle odpovídajícího modelu

4. Zobrazení informací do připravených textových polí

Alternativa: Nenalezen odpovídající model

V kroku 2 program nenalezne odpovídající model, který by korespondoval se strukturou předložené webové stránky a uživateli dává možnost vložit novou webovou stránku.

Alternativa: Nenalezeny data po extrakci

V kroku 4 program nevyextrahuje žádné informace a zobrazí tak prázdná textová pole.

5) Procházení výsledků

Aktér: Uživatel

Rozsah: Desktopová aplikace pro extrahování dat

Úroveň: Uživatelská

Úspěch: Zobrazení extrahovaných dat

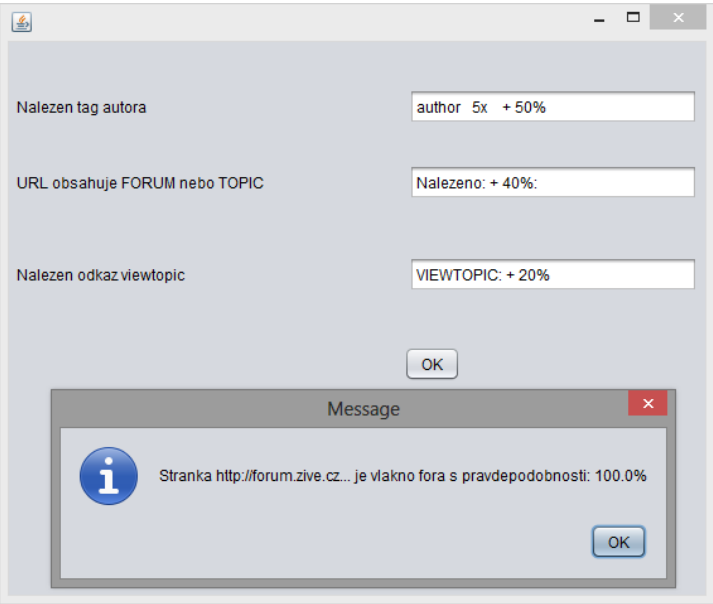
Spouštěč: Spustitelný soubor s koncovkou *.jar

Hlavní scénář:

1. Program zobrazí extrahované informace do připravených polí
2. Uživatel prohlíží data o úvodním příspěvku
3. Uživatel má možnost vybírat autory, které program extrahoval jako autory odpovědí na úvodní příspěvek
4. Program na základě vybraného autora odpovědi zobrazí text odpovědi

Příloha II. – Statistika vyhodnocení webové stránky

Ukázka vyhodnocovacího okna programu, kde jsou zobrazeny jednotlivé políčka s procentuálním ohodnocením.



Obrázek 27 – Výsledek testování webové stránky

První políčko na Obrázek 27 nám zobrazuje informaci o tom, kolikrát byl nalezen název třídy obsahující některé z klíčových slov uvedených v externím souboru. V našem případě se jedná o číslo 5 a uvnitř tagu, nesoucí název třídy bylo nalezeno klíčové slovo „author“ což nám podle definice správně připočetlo 50% k pravděpodobnosti, že se opravdu jedná o webové diskuzní fórum. Další políčko zobrazuje pouze skutečnost, zda URL odkazu obsahuje klíčové slovo „forum“ nebo „topic“, či nikoliv. Zde máme pozitivní nález, takže dalších 40%. V tuhle chvíli by nám pro vyhodnocení, zda se opravdu jedná o webové diskuzní fórum, stačily tyhle dvě podmínky. Ovšem pro úplnost se testují všechny tři a tudíž i existence odkazu vlákna na samo sebe ve zdrojovém kódu. Pokud sečteme pravděpodobnosti, výsledek nám dá více než 100% což je samozřejmě nesmysl. Pokud taková situace nastane, je pouze horní hranice nastavena na maximální hodnotu, a to 100%.